1

United States Patent [19]
Hellman et al.
[11] 4,200,770
[45] Apr. 29, 1980

[54] CRYPTOGRAPHIC APPARATUS AND METHOD

[75] Inventors: Martin E. Hellman, Stanford; Bailey W. Diffie, Berkeley; Ralph C. Merkle, Palo Alto, all of Calif.

[73] Assignee: Stanford University, Palo Alto, Calif.

[21] Appl. No.: 830,754

[22] Filed: Sep. 6, 1977

[51] Int. Cl.² H04L 9/04
[52] U.S. Cl. .... 178/22; 340/149 R; 375/2; 455/26
[58] Field of Search .... 178/22; 340/149 R

[56] References Cited

PUBLICATIONS

"New Directions in Cryptography", Diffie et al., *IEEE Transactions on Information Theory*, vol. IT-22, No. 6, Nov. 1976.

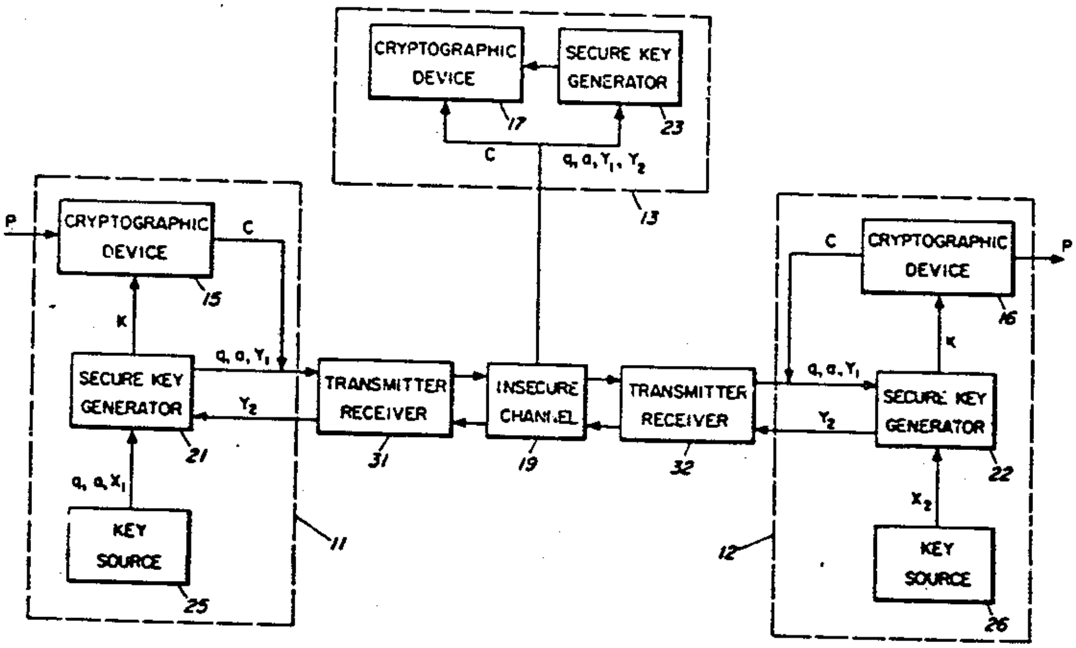
Diffie & Hellman, Multi-User Cryptographic Techniques", *AFIPS Conference Proceedings*, vol. 45, pp. 109-112, Jun. 8, 1976.

Primary Examiner—Howard A. Birmiel
Attorney, Agent, or Firm—Flehr, Hohbach, Test

[57] ABSTRACT

A cryptographic system transmits a computationally secure cryptogram over an insecure communication channel without prearrangement of a cipher key. A secure cipher key is generated by the conversers from transformations of exchanged transformed signals. The conversers each possess a secret signal and exchange an initial transformation of the secret signal with the other converser. The received transformation of the other converser's secret signal is again transformed with the receiving converser's secret signal to generate a secure cipher key. The transformations use non-secret operations that are easily performed but extremely difficult to invert. It is infeasible for an eavesdropper to invert the initial transformation to obtain either conversers' secret signal, or duplicate the latter transformation to obtain the secure cipher key.

8 Claims, 6 Drawing Figures



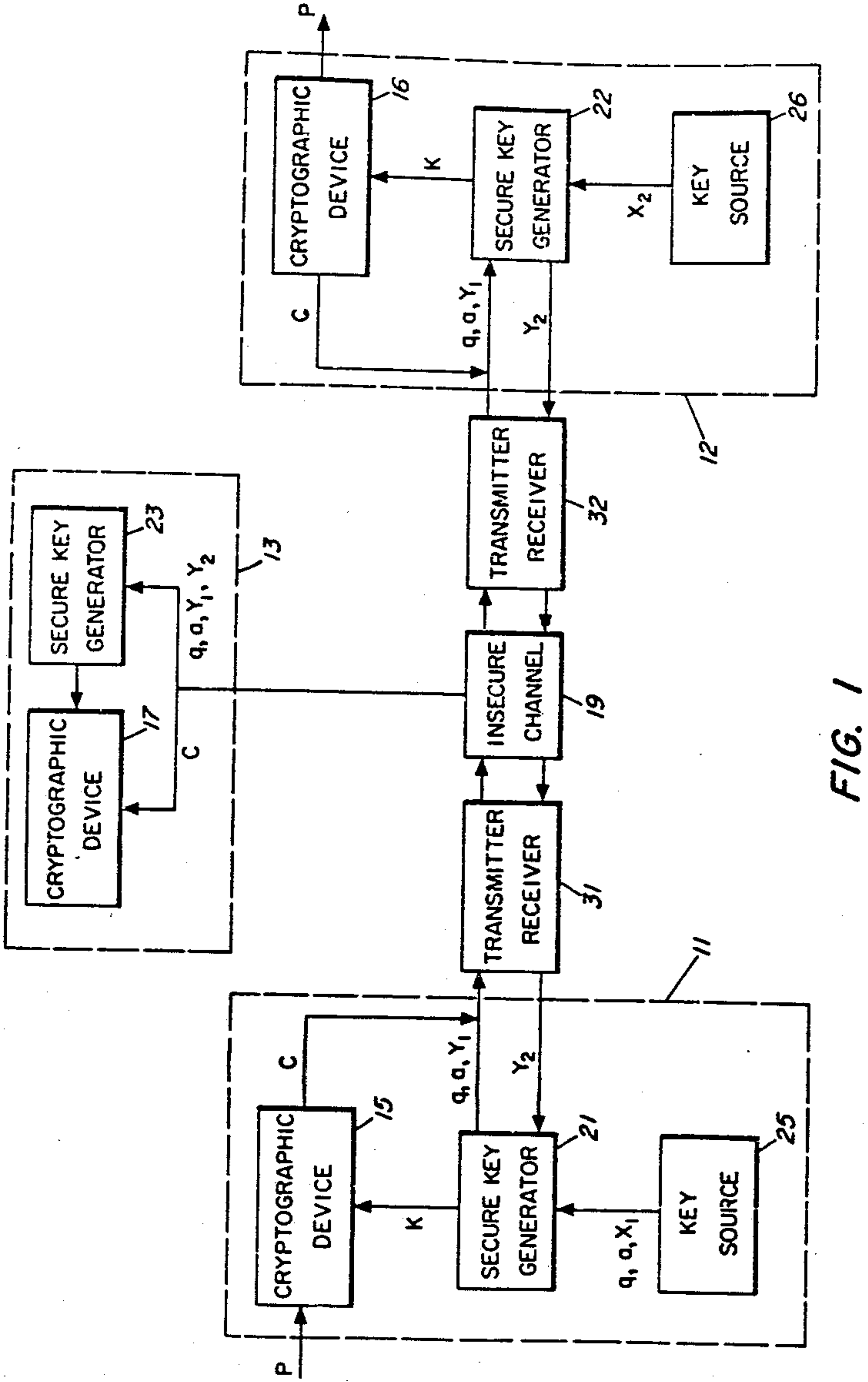

P
CRYPTOGRAPHIC DEVICE
16
K
SECURE KEY GENERATOR
22
X_2
KEY SOURCE
26
C
q, a, Y_1
Y_2
12
TRANSMITTER RECEIVER
32
INSECURE CHANNEL
19
TRANSMITTER RECEIVER
31
11
13
SECURE KEY GENERATOR
23
q, a, Y_1, Y_2
CRYPTOGRAPHIC DEVICE
17
C
C
q, a, Y_1
Y_2
CRYPTOGRAPHIC DEVICE
15
K
SECURE KEY GENERATOR
21
q, a, X_1
KEY SOURCE
25
P

FIG. 1

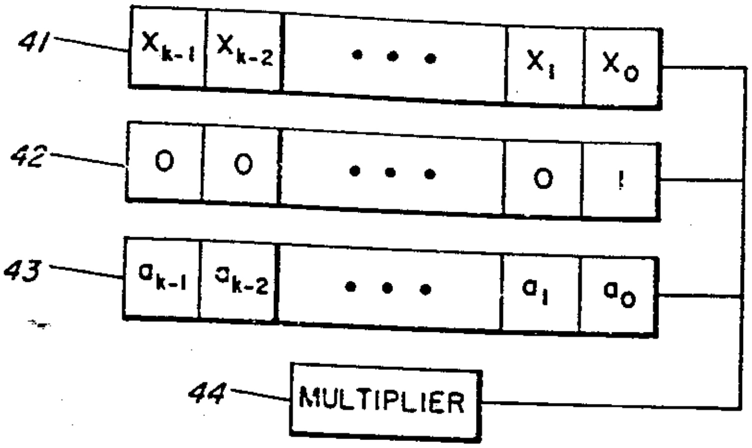

41
X_{k-1} X_{k-2} • • • X_1 X_0
42
0 0 • • • 0 1
43
a_{k-1} a_{k-2} • • • a_1 a_0
44
MULTIPLIER

FIG. 2

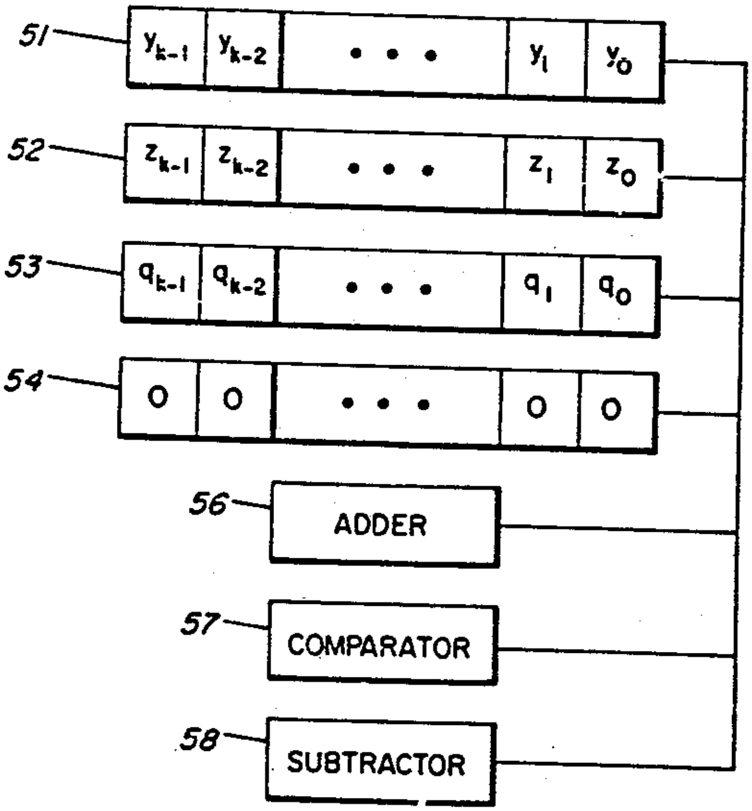

51
y_{k-1} y_{k-2} • • • y_1 y_0
52
z_{k-1} z_{k-2} • • • z_1 z_0
53
q_{k-1} q_{k-2} • • • q_1 q_0
54
0 0 • • • 0 0
56
ADDER
57
COMPARATOR
58
SUBTRACTOR

FIG. 3

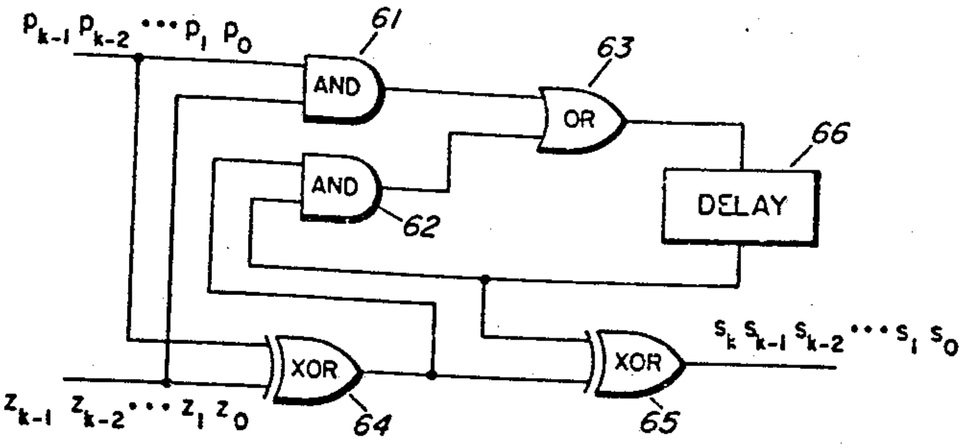

$p_{k-1}\ p_{k-2} \cdots p_1\ p_0$
61
AND
63
OR
66
AND
62
DELAY
XOR
XOR
$s_k\ s_{k-1}\ s_{k-2} \cdots s_1\ s_0$
$z_{k-1}\ z_{k-2} \cdots z_1\ z_0$
64
65

FIG. 4

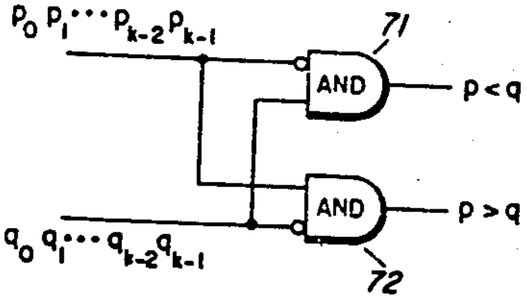

$p_0\ p_1 \cdots p_{k-2} p_{k-1}$
71
AND
p < q
AND
p > q
$q_0\ q_1 \cdots q_{k-2} q_{k-1}$
72

FIG. 5

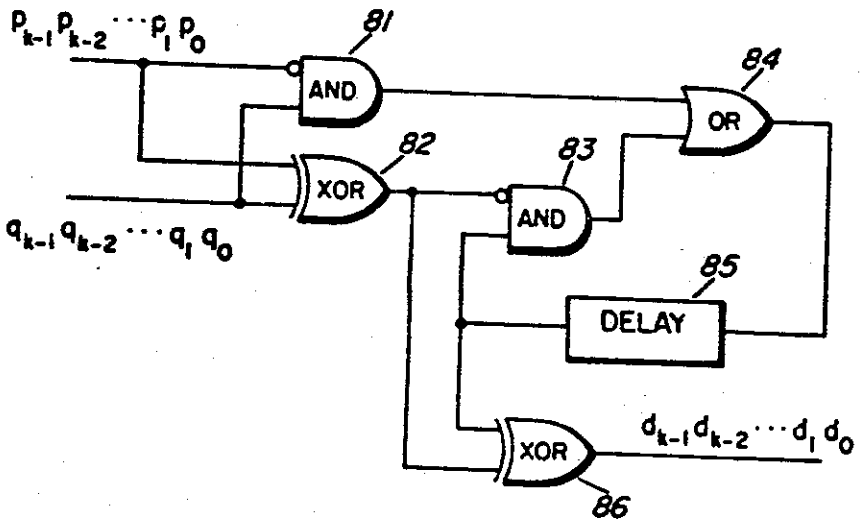

$p_{k-1} p_{k-2} \cdots p_1\ p_0$
81
AND
84
OR
82
83
XOR
AND
$q_{k-1} q_{k-2} \cdots q_1\ q_0$
85
DELAY
XOR
$d_{k-1} d_{k-2} \cdots d_1\ d_0$
86

FIG. 6

CRYPTOGRAPHIC APPARATUS AND METHOD

The Government has rights in this invention pursuant to Grant No. ENG-10173 of the National Science Foundation and IPA No. 0005.

BACKGROUND OF THE INVENTION

1. Field of Invention

The invention relates to cryptographic systems.

2. Description of Prior Art

Cryptographic systems are widely used to ensure the privacy and authenticity of messages communicated over insecure channels. A privacy system prevents the extraction of information by unauthorized parties from messages transmitted over an insecure channel, thus assuring the sender of a message that it is being read only by the intended receiver. An authentication system prevents the unauthorized injection of messages into an insecure channel, assuring the receiver of the message of the legitimacy of its sender.

One of the principal difficulties with existing cryptographic systems is the need for the sender and receiver to exchange a cipher key over a secure channel to which the unauthorized party does not have access. The exchange of a cipher key frequently is done by sending the key in advance over a secure channel such as private courier or registered mail; such secure channels are usually slow and expensive.

Diffie, et al, in "Multiuser Cryptographic Techniques," *AFIPS—Conference Proceedings*, Vol. 45, pp. 109-112, June 8, 1976, propose the concept of a public key cryptosystem that would eliminate the need for a secure channel by making the sender's keying information public. It is also proposed how such a public key cryptosystem could allow an authentication system which generates an unforgeable message dependent digital signature. Diffie presents the idea of using a pair of keys E and D, for enciphering and deciphering a message, such that E is public information while D is kept secret by the intended receiver. Further, although D is determined by E, it is infeasible to compute D from E. Diffie suggests the plausibility of designing such a public key cryptosystem that would allow a user to encipher a message and send it to the intended receiver, but only the intended receiver could decipher it. While suggesting the plausibility of designing such systems, Diffie presents neither proof that public key cryptosystems exist, nor a demonstration system.

Diffie suggests three plausibility arguments for the existence of a public key cryptosystem; a matrix approach, a machine language approach and a logic mapping approach. While the matrix approach can be designed with matrices that require a demonstrably infeasible cryptanalytic time (i.e., computing D from E) using known methods, the matrix approach exhibits a lack of practical utility because of the enormous dimensions of the required matrices. The machine language approach and logic mapping approach are also suggested, but there is not way shown to design them in such a manner that they would require demonstrably infeasible cryptanalytic time.

Diffie also introduces a procedure using the proposed public key cryptosystems, that could allow the receiver to easily verify the authenticity of a message, but which prevents him from generating apparently authenticated messages. Diffie describes a protocol to be followed to obtain authentication with the proposed public key cryptosystem. However, the authentication procedure relies on the existence of a public key cryptosystem which Diffie did not provide.

SUMMARY AND OBJECTS OF THE INVENTION

Accordingly, it is an object of this invention to allow authorized parties to a conversation (conversers) to converse privately even though an unauthorized party (eavesdropper) intercepts all of their communications.

Another object of this invention is to allow conversers on an insecure channel to authenticate each other's identity.

An illustrated embodiment of the present invention describes a method for communicating securely over an insecure channel without prearrangement of a cipher key. A secure cipher key is generated from transformations of exchanged transformed signals, which exchanged transformed signals are easy to effect but difficult to invert. The generated secure cipher key is used to encipher and decipher messages transmitted over the insecure communication channel.

This illustrated embodiment of the present invention describes a method and apparatus for generating a secure cipher key for use with conventional cryptographic communication between conversers over an insecure channel. The illustrated embodiment differs from a public key cryptosystem in that it provides a secure cipher key that is used with a conventional cryptographic system; a public key cryptosystem does not require a conventional cryptographic system. Further, the illustrated embodiment provides a means of transforming a signal that is practical to implement and is demonstrably infeasible to invert using known methods.

In the present invention a first converser transforms, in a manner infeasible to invert, a first signal while a second converser transforms, also in a manner infeasible to invert, a second signal. The first converser transmits the transformed first signal to the second converser, keeping the first signal secret, and the second converser transmits the transformed second signal to the first converser, keeping the second signal secret. The first converser then transforms the first signal with the transformed second signal to generate a third signal, representing a secure cipher key, that is infeasible to generate solely by transforming the transformed first signal and the transformed second signal. And, the second converser transforms the second signal with the transformed first signal to generate a fourth signal, also representing the secure cipher key, that is infeasible to generate solely by transforming the transformed first signal and the transformed second signal.

Another illustrated embodiment of the present invention describes a method for allowing a converser to authenticate another converser's identity. A first converser transforms, in a manner infeasible to invert, a first signal while a second converser transforms, also in a manner infeasible to invert, a second signal. The second converser places the transformed second signal in a public directory as the second converser's means of identification, keeping the second signal secret. The first converser transmits the transformed first signal to the second converser, with whom the first converser desires to communicate, keeping the first signal secret. The first converser then transforms the first signal with the second converser's transformed second signal, obtained from the public directory, to generate a third signal. The third signal represents a secure cipher key to be used for conventional cryptographic communication with the second converser, that is infeasible to generate solely by transforming the transformed first signal and the transformed second signal. The second converser transforms the second signal with the transformed first signal to generate a fourth signal, also representing the secure cipher key, that is infeasible to generate solely by transforming the transformed first signal and the transformed second signal. The second converser's identity is authenticated by the first converser by the second converser's ability to generate the fourth signal, representing the secure cipher key, and to use the secure cipher key in communicating over a conventional cryptographic system.

Additional objects and features of the present invention will appear from the description that follows wherein the preferred embodiments have been set forth in detail in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a cryptographic system that transmits a computationally secure cryptogram over an insecure communication channel.

FIG. 2 is a block diagram of a secure key generator for raising various numbers to various powers in modulo arithmetic.

FIG. 3 is a block diagram of a multiplier for performing multiplications in the secure key generator of FIG. 2.

FIG. 4 is a detailed schematic diagram of an adder for performing additions in the multiplier of FIG. 3.

FIG. 5 is a detailed schematic diagram of a comparator for performing magnitude comparisons in the multiplier of FIG. 3.

FIG. 6 is a detailed schematic diagram of a subtractor for performing subtractions in the multiplier of FIG. 3.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 1, a cryptographic system is shown in which all communications take place over an insecure communication channel 19, for example a telephone line. Two-way communication is exchanged on the insecure channel 19 between converser 11 and converser 12 using transmitter/receivers 31 and 32, for example modems sch as Bell 210 modems. Converser 11 possesses an unenciphered or plaintext message P to be communicated to converser 12. Converser 11 and converser 12 include cryptographic devices 15 and 16 respectively, for enciphering and deciphering information under the action of a cipher key K on line K. For example, the cryptographic devices 15 and 16 may include the recently adopted National Data Encryption Standard. The cryptographic devices 15 and 16 implement transformations S_K and S_K^{-1} (the transformation which is the inverse of S_K) when loaded with key K. For example, key K may be a sequence of random letters or digits. The cryptographic device 15 enciphers the plaintext message P into an enciphered message or ciphertext C on line C that is transmitted by converser 11 through the insecure channel 19; the ciphertext C is received by converser 12 and deciphered by cryptographic device 16 to obtain the plaintext message P. An unauthorized party or eavesdropper 13 is assumed to have a cryptographic device 17 and to have access to the insecure channel 19, so if he knew the key K he could decipher the ciphertext C to obtain the plaintext message P.

Converser 11 and converser 12 include independent key sources 25 and 26 respectively, which generate numbers or signals that represent numbers. For example, the key sources may be random number generators that are implemented from noisy amplifiers (e.g., Fairchild μ709 operational amplifiers) with a polarity detector. Key source 25 generates three signals, q, a, and X_1, and key source 26 generates X_2; a, X_1 and X_2 may be signals that represent independent random numbers chosen uniformly from the set of integers (1, 2, . . . $q-1$). Signals q and a are transmitted to the secure key generator 21 and are transmitted through the insecure channel 19 to secure key generator 22. Signals X_1 and X_2 are kept secret by converser 11 and converser 12 respectively, are given to the secure key generators 21 and 22 respectively, but are not transmitted through the insecure channel 19.

Converser 11 and converser 12 also include secure key generators 21 and 22 respectively, which accept the signals generated by the respective key sources 25 and 26. Secure key generator 22 also receives the signals q and a which are transmitted through the insecure channel 19. The secure key generators 21 and 22 generate signals Y_1 and Y_2 respectively by transforming X_1 and X_2 respectively with signals q and a in a manner that is easily performed but extremely difficult or infeasible to invert. A task is considered infeasible if its cost as measured by either the amount of memory used or the computing time is finite but impossibly large, for example, on the order of approximately 10^{30} operations with existing computational methods and equipment such transformations are characterized by the class of mathematical functions known as one-way cipher functions.

Signal Y_1 may be generated to represent the number obtained by raising the number represented by signal a to the power represented by signal X_1, modulo the number represented by signal q; this transformation may be represented symbolically as $Y_1 = a^{X_1} \bmod q$. Signal Y_2 may be generated to represent the number obtained by raising the number represented by signal a to the power represented by signal X_2, modulo the number represented by signal q; this transformation may be represented symbolically as $Y_2 = a^{X_2} \bmod q$.

Signals Y_1 and Y_2 are exchanged by transmitting Y_1 and Y_2 through the insecure channel 19 to secure key generators 22 and 21 respectively. Secure key generator 21 then generates a secure key K by transforming signal Y_2 with signals q, a and X_1, and secure key generator 22 generates the same secure key K by transforming Y_1 with signals q, a and X_2.

Secure key generator 21 may generate a secure key K represented by the number obtained by raising the number represented by signal Y_2 to the power represented by signal X_1, modulo the number represented by signal q; this transformation may be represented symbolically as

$$K = Y_2^{X_1} \bmod q = (a^{X_2})^{X_1} \bmod q = a^{X_1 X_2} \bmod q.$$

Secure key generator 22 may also generate the same secure key K represented by the number obtained by raising the number represented by signal Y_1 to the power represented by signal X_2, modulo the number represented by signal q; this transformation may be represented symbolically as

$$K = Y_1^{X_2} \bmod q = (a^{X_1})^{X_2} \bmod q = a^{X_1 X_2} \bmod q.$$

Conversers 11 and 12 then have the same secure key K which may be used with cryptographic devices 15 and 16.

The eavesdropper 13 is assumed to have a secure key generator 23 and to have access to all signals transmitted through the insecure channel 19, including signals q, a, Y_1, and Y_2. The difficulty of inverting the transformations which generated signals Y_1 and Y_2 make it infeasible for the eavesdropper 13 to generate signals X_1 or X_2. Further, the secure key K is infeasible to generate solely with signals q, a, Y_1 and Y_2.

The eavesdropper 13 is unabie to compute the secure key K by multiplication or exponentiation; multiplication yields

$$Y_1Y_2=a^{X_1+X_2} \bmod q \neq K$$

and exponentiation yields either

$$Y_1{}^{Y_2}=a^{(X_1a^{X_1})}\neq K$$

or

$$Y_2{}^{Y_1}=a^{(X_2a^{X_1})}\neq K.$$

The eavesdropper in theory could obtain X_1 or X_2 from q, a and Y_1 and Y_2 by raising a to the first, second, third, etc., powers until Y_1 or Y_2 was obtained. This is prevented by choosing q to be a large number; if q is a 200 bit quantity, the average number of trials before success is on the order of $2^{198}=4\times10^{59}$ and is physically infeasible. Improved algorithms for computing logarithms modulo q (if $Y=a^X$ mod q, X is the logarithm of Y to the base a modulo q) are known but, if $q=2r+1$ with q and r being prime, then the most efficient known algorithm requires approximately $q^{\frac{1}{2}}$ operations. Again, taking $q=2^{200}$, about $2^{100}=10^{30}$ operations are required, still physically infeasible. An example of such a paid is $r=(2^{121}\cdot5^2\cdot7^2\cdot11^2\cdot13\cdot17\cdot19\cdot23\cdot29\cdot31\cdot37\cdot41\cdot43\cdot47\cdot53\cdot59)+1$ and $q=2r+1$. Other restrictions on q, a, X_1 and X_2 may also be imposed.

The secure key generators 21 and 22, for raising various numbers to various powers modulo q, can be implemented in electronic circuitry as shown in FIG. 2. For ease of illustration, FIG. 2 depicts raising a to the X power modulo q; raising Y to the X power modulo q is obtained by initially loading Y, instead of a, into the A register 43.

FIG. 2 shows the initial contents of three registers 41, 42 and 43. The binary representation of X ($x_{k-1}x_{k-2}\ldots x_1x_0$) is loaded into the X register 41; 1 is loaded into the R register 42; and, the binary representation of a is loaded into the A register 43, corresponding to i=0. The number of bits k in each register is the least integer such that $2^k \geqq q$. If k=200, then all three registers can be obtained from a single 1024 bit random access memory (RAM) such as the Intel 2102. The implementation of multiplier 44, for multiplying two numbers modulo q, will be described in more detail later.

Referring to FIG. 2, if the low order bit, containing x_0, of the X register 41 equals 1 then the R register 42 and the A register 43 contents are multiplied modulo q and their product, also a k bit quantity, replaces the contents of the R register 42. If $x_0=0$, the R register 42 contents are left unchanged. In either case, the A register 43 is then loaded twice into the multiplier 44 so that the square, modulo q, of the A register 43 contents is computed. This value, $a^{(2^i+1)}$, replaces the contents of the A register 43. The X register 41 contents are shifted one bit to the right and a 0 is shifted in at the left so its contents are now $0x_{k-1}x_{k-2}\ldots x_2x_1$.

The low order bit, containing x_1, of the X register 41 is examined. If it equals 1 then, as before, the R register 42 and A register 43 contents are multiplied modulo q and their product replaces the contents of the R register 42. If the low order bit equals 0 then the R register 42 contents are left unchanged. In either case, the contents of the A register 43 are replaced by the square, modulo q, of the previous contents. The X register 41 contents are shifted one bit to the right and a 0 is shifted in at the left so its contents are now $00x_{k-1}x_{k-2}\ldots x_3x_2$.

This process continues until the X register 41 contains all 0's, at which point the value of a^X modulo q is stored in the R register 42.

An example is helpful in following this process. Taking q=23, we find k=5 from $2^k \geqq q$. If a=7 and X=18 then $a^X=7^{18}=1{,}628{,}413{,}597{,}910{,}449=23(70{,}800{,}591{,}213{,}497)+18$ so a^X modulo q equals 18. This straightforward but laborious method of computing a^X modulo q is used as a check to show that the method of FIG. 2, shown below, yields the correct result. The R register 42 and A register 43 contents are shown in decimal form to facilitate understanding.

i	X (in binary)	R	A
0	10010	1	7
1	01001	1	3
2	00100	3	9
3	00010	3	12
4	00001	3	6
5	00000	18	13

The row marked i=0 corresponds to the initial contents of each register, X=18, R=1 and A=a=7. Then, as described above, because the right most bit of X register 41 is 0, the R register 42 contents are left unchanged, the contents of the A register 43 are replaced by the square, modulo 23, of its previous contents ($7^2=49=2\times23+3=3$ modulo 23), the contents of the X register 41 are shifted one bit to the right, and the process continues. Only when i=1 and 4 do the rightmost bit of the X register 41 contents equal 1, so only going from i=1 to 2 and from i=4 to 5 is the R register 42 replaced by RA modulo q. When i=5, X=0 so the process is complete and the result, 18, is in the R register 42.

Note that the same result, 18, is obtained here as in the straightforward calculation of 7^{18} modulo 23, but that here large numbers never resulted.

Another way to understand the process is to note that the A register contains a, a^2, a^4, a^8 and a^{16} when i=0, 1, 2, 3 and 4 respectively, and that $a^{18}=a^{16}\,a^2$, so only these two values need to be multiplied.

FIG. 3 continues the description of this illustrative implementation by depicting an implementation of the modulo q multiplier 44 in FIG. 2. The two numbers, y and z, to be multiplied are loaded into the Y and Z registers 51 and 52 respectively, and q is loaded in the Q register 53. The product yz modulo q will be produced in the P register 54 which is initially set to 0. If k=200, then all four registers can be obtained from a single 1024 bit RAM such as the Intel 2102. The implementation of FIG. 3 is based on the fact that y z mod q=y_0z mod q+$2y_1z$ mod q+$4y_2z$ mod q+ . . . +$2^{k-1}y_{k-1}z$ mod q, where $y_{k-1}y_{k-2} \ldots y_1y_0$ is the binary representation of Y.

To multiply y times z, if the rightmost bit, containing y_0, of the Y register 51 is 1 then the contents of the Z register 53 are added to the P register 54 by adder 55. If $y_0=0$, then the P register 54 is unchanged. Then the Q and P register contents are compared by comparator 56 to determine if the contents of the P register 54 are greater than or equal to q, the contents of the Q register 53. If the contents of the P register 54 are greater than or equal to q then subtractor 57 substracts q from the contents of the P register 54 and places the difference in the P register 54, if less than q the P register 54 is unchanged.

Next, the contents of Y register 51 are shifted one bit to the right and a 0 is shifted in at the left so its contents become $0y_{k-1}\,y_{k-2} \ldots y_2y_1$, so that y_1 is ready for computing $2y_1z$ mod q. The quantity 2z mod q is computed for this purpose by using adder 55 to add z to itself, using comparator 56 to determine if the result, 2z, is less than q, and using subtractor 57 for subtracting q from 2z if the result is not less than q. The result, 2z mod q is then stored in the Z register 52. The rightmost bit, containing y_1, of the Y register 51 is then examined, as before, and the process repeats.

This process is repeated a maximum of k times or until the Y register 51 contains all 0's, at which point xy modulo q is stored in the P register 54.

As an example of these operations, consider the problem of computing 7×7 modulo 23 needed to produce the second state of the A register when 7^{18} mod 23 was computed. The following steps show the successive contents of the Y, Z and P registers which result in the answer $7\times7=3$ modulo 23.

i	Y (in binary)	Z	P
0	00111	7	0
1	00011	14	0 + 7 = 7
2	00001	5	7 + 14 = 21
3	00000	10	21 + 5 = 3 mod 23

FIG. 4 depicts an implementation of an adder 55 for adding two k bit numbers p and z. The numbers are presented one bit at a time to the device, low order bit first, and the delay element is initially set to 0. (The delay represents the binary carry bit.) The AND gate 61 determines if the carry bit should be a one based on p_i and z_i both being 1 and the AND gate 62 determines if the carry should be a 1 based on the previous carry being a 1 and one of p_i or z_i being 1. If either of these two conditions is met, the OR gate 63 has an output of 1 indicating a carry to the next stage. The two exclusive-or (XOR) gates 64 and 65 determine the i^{th} bit of the sum, s_i, as the modulo-2 sum of p_i, z_i and the carry bit from the previous stage. The delay 66 stores the previous carry bit. Typical parts for implementing these gates and the delay are SN7400, SN7404, and SN7474.

FIG. 5 depicts an implementation of a comparator 56 for comparing two numbers p and q. The two numbers are presented one bit at a time, high order bit first. If neither the p<q nor the p>q outputs have been triggered after the last bits p_0 and q_0 have been presented, then p=q. The first triggering of either the p<q or the p>q output causes the comparison operation to cease. The two AND gates 71 and 72 each have one input inverted (denoted by a circle at the input). An SN7400 and SN7404 provide all of the needed logic circuits.

FIG. 6 depicts an implementation of a subtractor 57 for subtracting two numbers. Because the numbers subtracted in FIG. 3 always produce a non-negative difference, there is no need to worry about negative differences. The larger number, the minuend, is labelled p and the smaller number, the substrahend, is labelled q. Both p and q are presented serially to the subtractor 57, low order bit first. AND gates 81 and 83, OR gate 84 and XOR gate 82 determine if borrowing (negative carrying) is in effect. A borrow occurs if either $p_i=0$ and $q_i=1$, or $p_i=q_i$ and borrowing occurred in the previous stage. The delay 85 stores the previous borrow state. The i^{th} bit of the difference, d_i, is computed as the XOR, or modulo-2 difference, of p_i, q_i and the borrow bit. The output of XOR gate 82 gives the modulo-2 difference between p_i and q_i, and XOR gate 86 takes the modulo-2 difference of this with the previous borrow bit. Typical parts for implementing these gates and the delay are SN7400, SN7404 and SN7474.

There are many methods for implementing this form of the invention. The signals q and a may be public knowledge rather than generated by the key source 25. Further, it should be appreciated that the present invention has the capability of being modified by the use of additional transformations or exchanges of signals.

In some applications, it will prove valuable to have the i^{th} converser on the system generate Y_i as above and place it in a public file or directory rather than transmitting it to another converser with whom he wishes to communicate. Then two conversers i and j who wish to establish a secure channel will use $K_{ij}=Y_i^{X_j}$ mod $q=Y_j^{X_i}$ mod q as their key. The advantage is that converser i, having once proved his identity to the system through the use of his driver's license, fingerprint, etc., can prove his identity to converser j by his ability to compute K_{ij} and encrypt data with it.

Variations on the above described embodiment are possible. For example, in the above method based on logarithms modulo q, m-dimensional vectors, each of whose components are between 0 and q−1 could also be used. Then all operations are performed in the finite field with q^m elements, which operations are well described in the literature. Thus, although the best mode contemplated for carrying out the present invention has been herein shown and described, it will be apparent that modification and variation may be made without departing from what is regarded to be the subject matter of this invention.

What is claimed is:

1. A secure key generator comprising:

a first input connected to receive an applied first signal;

a second input connected to receive an applied second signal;

a first output;

a second output; and

means for generating at the first output a third signal, that is a transformation of said first signal and which transformation is infeasible to invert, and for generating at the second output a fourth signal, that is a transformation of said second signal with said first signal, which represents a secure key and is infeasible to generate solely with said second signal and said third signal.

2. In a method of communicating securely over an insecure communication channel of the type which communicates a message from a transmitter to a receiver, the improvement characterized by:

generating and transforming, in a manner infeasible to invert, a first signal at the transmitter to generate a transformed first signal;

generating and transforming, in a manner infeasible to invert, a second signal at the receiver to generate a transformed second signal;

transmitting said transformed first signal from the transmitter to the receiver;

transmitting said transformed second signal from the receiver to the transmitter;

transforming said transformed second signal with said first signal at the transmitter to generate a third signal, representing a secure cipher key, that is infeasible to generate solely with said transformed first signal and said transformed second signal;

transforming said transformed first signal with said second signal at the receiver to generate a fourth signal that is identical to the third signal and represents said secure cipher key;

enciphering the message with said secure cipher key at the transmitter;

transmitting the enciphered message from the transmitter to the receiver; and

deciphering the enciphered message with said secure cipher key at the receiver.

3. In a method of communicating securely over an insecure communication channel as in claim 2, further comprising:

authenticating the receiver's identity at the transmitter from the receiver's ability to generate the fourth signal, representing the secure cipher key.

4. A method of generating a secure cipher key between a transmitter and receiver comprising the steps of:

generating and transforming, in a manner infeasible to invert, a first signal at the transmitter to generate a transformed first signal;

generating and transforming, in a manner infeasible to invert, a second signal at the receiver to generate a transformed second signal;

transmitting said transformed first signal from the transmitter to the receiver

transmitting said transformed second signal from the receiver to the transmitter;

transforming said transformed second signal with said first signal at the transmitter to generate a third signal, representing a secure cipher key, that is infeasible to generate solely with said transformed first signal and said transformed second signal; and

transforming said transformed first signal with said second signal at the receiver to generate a fourth signal that is identical to the third signal and represents said secure cipher key.

5. An apparatus for generating a secure cipher key comprising:

a first secure key generator having a first input connected to receive an applied first signal, having a second input connected to receive a second signal, having a first and second outputs, and having a means for generating at the first output a third signal, that is a transformation of said first signal and which transformation is infeasible to invert, and for generating at the second output a fourth signal, that is a transformation of said second signal with said first signal, which represents a secure key and is infeasible to generate solely with said second signal and said third signal; and

a second secure key generator having a first input connected to receive an applied fifth signal, having a second input connected to receive said third signal, having a first and second outputs, and having a means for generating at the first output said second signal, that is a transformation of said fifth signal and which transformation is infeasible to invert, and for generating at the second output a sixth signal, that is a transformation of said third signal with said fifth signal, which represents the secure key and is infeasible to generate solely with said second signal and said third signal.

6. A method of generating a secure cipher key between a transmitter and receiver comprising the steps of:

transforming, in a manner infeasible to invert, a first signal at the transmitter to generate a transformed first signal wherein transforming said first signal is performed by raising a first number to a power represented by said first signal, modulo a second number;

transforming, in a manner infeasible to invert, a second signal at the receiver to generate a transformed second signal, wherein transforming said second signal is performed by raising the first number to a power represented by said second signal, modulo the second number;

transmitting said transformed first signal from the transmitter to the receiver;

transmitting said transformed second signal from the receiver to the transmitter;

transforming said transformed second signal with said first signal at the transmitter to generate a third signal, representing a secure cipher key, that is infeasible to generate solely with said transformed first signal and said transformed second signal, wherein transforming said transformed second signal with said first signal is performed by raising a number represented by said transformed second signal to a power represented by said first signal, modulo the second number; and

transforming said transformed first signal with said second signal at the receiver to generate a fourth signal, representing said secure cipher key, that is infeasible to generate solely with said transformed first signal and said transformed second signal, wherein transforming said transformed first signal with said second signal is performed by raising a number represented by said transformed first signal to a power represented by said second signal, modulo the second number.

7. An apparatus for generating a secure cipher key comprising:

a first secure key generator having a first input connected to receive an applied first signal, having a second input connected to receive a second signal, having first and second outputs, and having a means for generating at the first output a third signal, that is a transformation of said first signal in which said transformation includes raising a first number to a power represented by said first signal, modulo or second number, and for generating at the second output a fourth signal, that is a transformation of said second signal with said first signal which transformation includes raising a number represented by said second signal to a power represented by said first signal, modulo the second number, which represents a secure key and is infeasible to generate solely with said second signal and said third signal; and

a second secure key generator having a first input connected to receive an applied fifth signal, having a second input connected to receive said third signal, having a first and second outputs, and having a means for generating at the first output said second signal, that is a transformation of said fifth signal in which said transformation includes raising a first number to a power represented by said fifth signal, modulo the second number, and for generating at the second output a sixth signal, that is a transformation of a said third signal with said fifth signal which transformation includes raising a number represented by said third signal to a power represented by said fifth signal, modulo the second number, which represents the secure key and is infeasible to generate solely with said second signal and said third signal.

8. An apparatus for generating a secure cipher key comprising:

a first secure key generator having a first input connected to receive an applied first signal, having a second input connected to receive a second signal, having a first and second outputs, and having a means for generating at the first output a third signal, said third signal Y_i being described by

$$Y_i=a^{x_i} \bmod q$$

where

q=a large prime number

a=a random number, such that $1 \leqq a \leqq q-1$

x_i=the first signal which represents a random number, such that $1 \leqq X_i \leqq q-1$

a transformation of said first signal which is infeasible to invert, and for generating at the second output a fourth signal, said fourth signal K_{ij} being described by

$$K_{ij}=Y_j^{X_i} \bmod q$$

where

Y_j=the second signal

a transformation of said second signal with said first signal, which represents said secure cipher key and is infeasible to generate solely with said second signal and said third signal; and

a second secure key generator having a first input connected to receive an applied fifth signal, having a second input connected to receive said third signal, having a first and second outputs, and having a means for generating at the first output a second signal, said second signal Y_j being described by

$$Y_j=a^{X_j} \bmod q$$

where

X_j=the fifth signal which represents a random number, such that $1 \leqq X_j \leqq q-1$

a transformation of said fifth signal which is infeasible to invert, and for generating at the second output a sixth signal, said sixth signal K_{ij} being described by

$$K_{ij}=Y_i^{X_j} \bmod q$$

a transformation of said third signal with said fifth signal, which represents the secure key and is infeasible to generate solely with said second signal and said third signal.

* * * * *

2

RSA Data Security, Inc.

BSAFE

A Cryptographic Toolkit

User's Manual

RSA

DATA SECURITY INC

THE KEYS TO
PRIVACY AND
AUTHENTICATION

VERSION 2.1

When you ask for a particular individual's public key, the CA will send the certificate and, as a digital signature, the digest of the certificate encrypted with the CA's private key. To verify that the certificate is genuine, digest the certificate and decrypt the signature using the CA's public key. Compare the two results, if they are the same, you have a proper certificate.

If the CA you deal with does not have a certificate for the individual in question, that CA can talk to another CA which may have the right certificate. In fact, to find a particular certificate, a CA may have to go through a chain of CA's until it finds one that possesses the desired certificate.

Names that uniquely distinguish users are necessary for digital certificates to be of real use. The CCITT X.500 series of documents offer more discussion regarding naming conventions and related topics.

Diffie-Hellman Public Key Agreement

Whitfield Diffie and Martin Hellman invented this, the first true public-key algorithm, in 1976. It provides for key agreement, but not encryption or authentication.

The Diffie-Hellman key agreement algorithm provides a method for two parties to each compute the same secret key without exchanging secret information. Its security relies on the difficulty of computing discrete logarithms modulo a prime number. It takes very little time to exponentiate modulo a prime number, but much more to compute the inverse, the discrete logarithm. The RSA Laboratories publication, "Frequently Asked Questions About Today's Cryptography," declares, "The best discrete log problems have expected running times similar to that of the best factoring algorithms." That is, the time it takes to compute discrete logs modulo a prime of a certain length is about equivalent to the time it takes to factor a number of that same length. See the section titled "The RSA Algorithm" for a discussion on factoring.

The Diffie-Hellman algorithm is made up of three parts, Parameter Generation, Phase 1 and Phase 2.

Parameter Generation

A central authority selects a prime number p of length k bytes ($k \cdot 8$ bits), and an integer $g \in (0, p)$, called the base. The central authority may optionally select an integer l, the private value length in bits, that satisfies $2^{l-1} \leq p$.

Phase 1

Each of the two parties executing the Diffie-Hellman protocol randomly generates a private value $x \in (0, p-1)$, that is, a number greater than 0 but less than the prime. If the central authority had specified the length l, the private value shall satisfy $2^{l-1} \leq x < 2^{l}$.

Each party will then compute a public value $y = g^{x} \bmod p$. The two parties will exchange the public values.

These private and public values correspond to the private and public key components of a key pair. The public value is generated in such a way that computing the private value from the public number is computationally infeasible.

Phase 2

Each participant computes the agreed-upon secret key, z, from the other's public value, y', their own private value, x, and the prime, p: $z = (y')^{x} \bmod p$.

Even with knowledge of the parameters and both public keys, an outside individual will not be able to determine the secret key. One must have one of the private values to determine the secret key. This means secret information is never sent over unsecure lines.

The Math

Even though the two parties involved are making computations using different private values, they will both end up with the same secret key, as illustrated by the following.

p	: prime
g	: base
x_1	: 1st party private value
x_2	: 2nd party private value
y_1	: 1st party public value
y_2	: 2nd party public value
z	: secret key

Each party computes a private value, x_n, and then a public value, y_n, in phase 1.

$$y_1 = g^{x_1} \bmod p$$

$$y_2 = g^{x_2} \bmod p$$

In phase 2, the parties trade public values and compute the same secret key.

- $z = y_2^{x_1} \bmod p$

  $z = y_1^{x_2} \bmod p$

They both compute the same z, because

$$y_2^{x_1} = (g^{x_2})^{x_1} = (g^{x_1})^{x_2} = y_1^{x_2} \bmod p$$

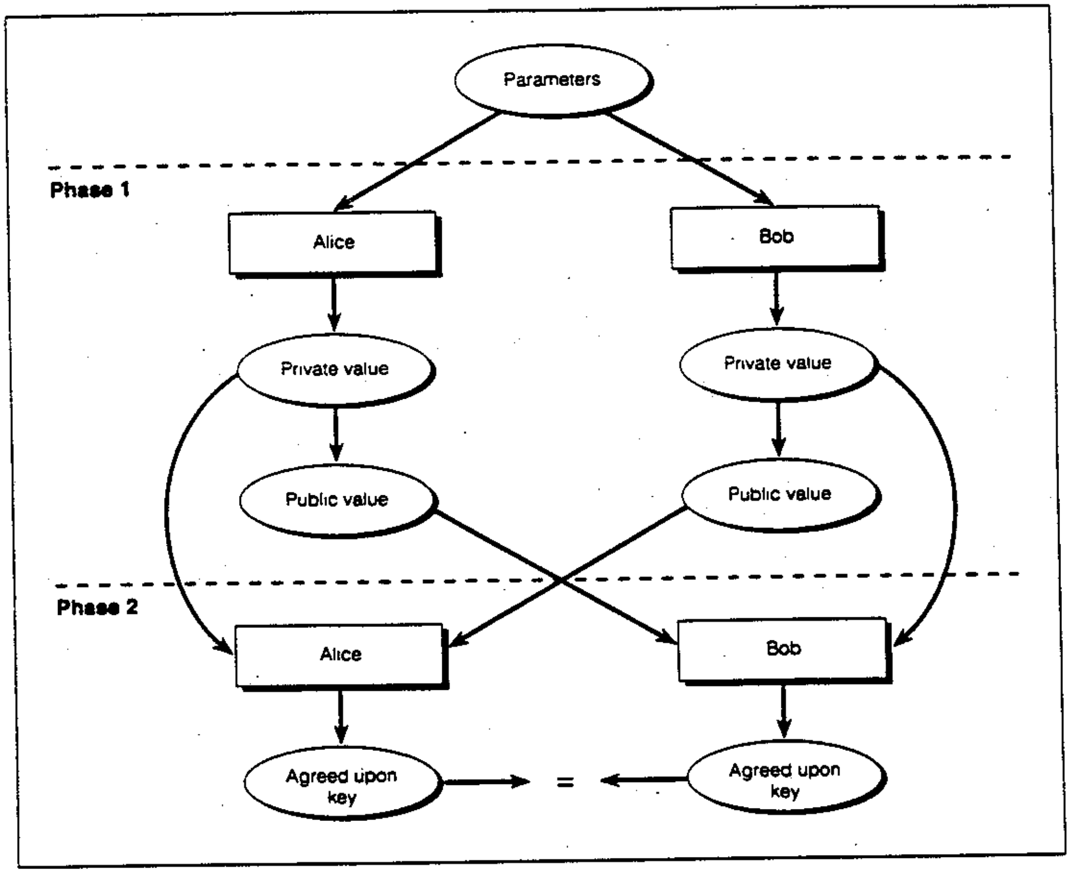


Figure 2.10 The Diffie-Hellman Key Agreement Protocol

The above protocol could be extended to more than two parties. For multiple parties each individual would choose a private value and use a collection of public values form other parties to generate a common secret key.

3

NEW DIRECTIONS IN CRYPTOGRAPHY

W. Diffie & M. Hellman

Reprinted from IEEE Transactions on Information Theory, November 1976

Printed in U.S.A. Annals No. 611IT001

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The development of computer controlled communication networks promises effortless and inexpensive contact between people or computers on opposite sides of the world, replacing most mail and many excursions with telecommunications. For many applications these contacts must be made secure against both eavesdropping and the injection of illegitimate messages. At present, however, the solution of security problems lags well behind other areas of communications technology. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits of teleprocessing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver is able to decipher it. As such, a public key cryptosystem is a multiple access cipher. A private conversation can therefore be held between any two individuals regardless of whether they have ever communicated before. Each one sends messages to the other enciphered in the receiver's public enciphering key and deciphers the messages he receives using his own secret deciphering key.

We propose some techniques for developing public key cryptosystems, but the problem is still largely open.

Public key distribution systems offer a different approach to eliminating the need for a secure key distribution channel. In such a system, two users who wish to exchange a key communicate back and forth until they arrive at a key in common. A third party eavesdropping on this exchange must find it computationally infeasible to compute the key from the information overheard. A possible solution to the public key distribution problem is given in Section III, and Merkle [1] has a partial solution of a different form.

A second problem, amenable to cryptographic solution, which stands in the way of replacing contemporary busi-

Manuscript received June 3, 1976. This work was partially supported by the National Science Foundation under NSF Grant ENG 10173. Portions of this work were presented at the IEEE Information Theory Workshop, Lenox, MA, June 23–25, 1975 and the IEEE International Symposium on Information Theory in Ronneby, Sweden, June 21–24, 1976.

W. Diffie is with the Department of Electrical Engineering, Stanford University, Stanford, CA, and the Stanford Artificial Intelligence Laboratory, Stanford, CA 94305.

M. E. Hellman is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

Printed in U.S.A. Annals No. 611IT001

ness communications by teleprocessing systems is authentication. In current business, the validity of contracts is guaranteed by signatures. A signed contract serves as legal evidence of an agreement which the holder can present in court if necessary. The use of signatures, however, requires the transmission and storage of written contracts. In order to have a purely digital replacement for this paper instrument, each user must be able to produce a message whose authenticity can be checked by anyone, but which could not have been produced by anyone else, even the recipient. Since only one person can originate messages but many people can receive messages, this can be viewed as a broadcast cipher. Current electronic authentication techniques cannot meet this need.

Section IV discusses the problem of providing a true, digital, message dependent signature. For reasons brought out there, we refer to this as the one-way authentication problem. Some partial solutions are given, and it is shown how any public key cryptosystem can be transformed into a one-way authentication system.

Section V will consider the interrelation of various cryptographic problems and introduce the even more difficult problem of trap doors.

At the same time that communications and computation have given rise to new cryptographic problems, their offspring, information theory, and the theory of computation have begun to supply tools for the solution of important problems in classical cryptography.

The search for unbreakable codes is one of the oldest themes of cryptographic research, but until this century all proposed systems have ultimately been broken. In the nineteen twenties, however, the "one time pad" was invented, and shown to be unbreakable [2, pp. 398–400]. The theoretical basis underlying this and related systems was put on a firm foundation a quarter century later by information theory [3]. One time pads require extremely long keys and are therefore prohibitively expensive in most applications.

In contrast, the security of most cryptographic systems resides in the computational difficulty to the cryptanalyst of discovering the plaintext without knowledge of the key. This problem falls within the domains of computational complexity and analysis of algorithms, two recent disciplines which study the difficulty of solving computational problems. Using the results of these theories, it may be possible to extend proofs of security to more useful classes of systems in the foreseeable future. Section VI explores this possibility.

Before proceeding to newer developments, we introduce terminology and define threat environments in the next section.

II. Conventional Cryptography

Cryptography is the study of "mathematical" systems for solving two kinds of security problems: privacy and authentication. A privacy system prevents the extraction of information by unauthorized parties from messages transmitted over a public channel, thus assuring the sender of a message that it is being read only by the intended recipient. An authentication system prevents the unauthorized injection of messages into a public channel, assuring the receiver of a message of the legitimacy of its sender.

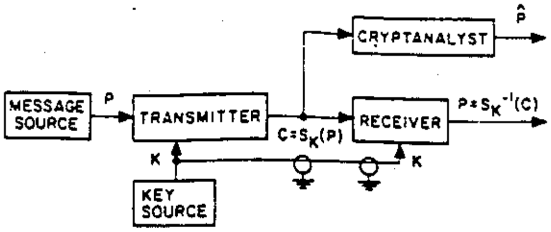


Fig. 1. Flow of information in conventional cryptographic system.

A channel is considered public if its security is inadequate for the needs of its users. A channel such as a telephone line may therefore be considered private by some users and public by others. Any channel may be threatened with eavesdropping or injection or both, depending on its use. In telephone communication, the threat of injection is paramount, since the called party cannot determine which phone is calling. Eavesdropping, which requires the use of a wiretap, is technically more difficult and legally hazardous. In radio, by comparison, the situation is reversed. Eavesdropping is passive and involves no legal hazard, while injection exposes the illegitimate transmitter to discovery and prosecution.

Having divided our problems into those of privacy and authentication we will sometimes further subdivide authentication into message authentication, which is the problem defined above, and user authentication, in which the only task of the system is to verify that an individual is who he claims to be. For example, the identity of an individual who presents a credit card must be verified, but there is no message which he wishes to transmit. In spite of this apparent absence of a message in user authentication, the two problems are largely equivalent. In user authentication, there is an implicit message "I AM USER X," while message authentication is just verification of the identity of the party sending the message. Differences in the threat environments and other aspects of these two subproblems, however, sometimes make it convenient to distinguish between them.

Fig. 1 illustrates the flow of information in a conventional cryptographic system used for privacy of communications. There are three parties: a transmitter, a receiver, and an eavesdropper. The transmitter generates a plaintext or unenciphered message P to be communicated over an insecure channel to the legitimate receiver. In order to prevent the eavesdropper from learning P, the transmitter operates on P with an invertible transformation S_K to produce the ciphertext or cryptogram $C = S_K(P)$. The key K is transmitted only to the legitimate receiver via a secure channel, indicated by a shielded path in Fig. 1. Since the legitimate receiver knows K, he can decipher C by operating with S_K^{-1} to obtain $S_K^{-1}(C) = S_K^{-1}(S_K(P)) = P$, the original plaintext message. The secure channel cannot be used to transmit P itself for reasons of capacity or delay. For example, the secure channel might be a weekly courier and the insecure channel a telephone line.

A *cryptographic system* is a single parameter family $\{S_K\}_{K \in \{K\}}$ of invertible transformations

$$S_K{:}\{P\} \to \{C\} \tag{1}$$

from a space $\{P\}$ of plaintext messages to a space $\{C\}$ of ciphertext messages. The parameter K is called the key and is selected from a finite set $\{K\}$ called the keyspace. If the message spaces $\{P\}$ and $\{C\}$ are equal, we will denote them both by $\{M\}$. When discussing individual cryptographic transformations S_K, we will sometimes omit mention of the system and merely refer to the transformation K.

The goal in designing the cryptosystem $\{S_K\}$ is to make the enciphering and deciphering operations inexpensive, but to ensure that any successful cryptanalytic operation is too complex to be economical. There are two approaches to this problem. A system which is secure due to the computational cost of cryptanalysis, but which would succumb to an attack with unlimited computation, is called *computationally secure*; while a system which can resist any cryptanalytic attack, no matter how much computation is allowed, is called *unconditionally secure*. Unconditionally secure systems are discussed in [3] and [4] and belong to that portion of information theory, called the Shannon theory, which is concerned with optimal performance obtainable with unlimited computation.

Unconditional security results from the existence of multiple meaningful solutions to a cryptogram. For example, the simple substitution cryptogram *XMD* resulting from English text can represent the plaintext messages: now, and, the, etc. A computationally secure cryptogram, in contrast, contains sufficient information to uniquely determine the plaintext and the key. Its security resides solely in the cost of computing them.

The only unconditionally secure system in common use is the *one time pad*, in which the plaintext is combined with a randomly chosen key of the same length. While such a system is provably secure, the large amount of key required makes it impractical for most applications. Except as otherwise noted, this paper deals with computationally secure systems since these are more generally applicable. When we talk about the need to develop provably secure cryptosystems we exclude those, such as the one time pad, which are unwieldy to use. Rather, we have in mind systems using only a few hundred bits of key and implementable in either a small amount of digital hardware or a few hundred lines of software.

We will call a task *computationally infeasible* if its cost as measured by either the amount of memory used or the runtime is finite but impossibly large.

Much as error correcting codes are divided into convolutional and block codes, cryptographic systems can be divided into two broad classes: *stream ciphers* and *block ciphers*. Stream ciphers process the plaintext in small chunks (bits or characters), usually producing a pseudorandom sequence of bits which is added modulo 2 to the bits of the plaintext. Block ciphers act in a purely combinatorial fashion on large blocks of text, in such a way that a small change in the input block produces a major change in the resulting output. This paper deals primarily with block ciphers, because this *error propagation* property is valuable in many authentication applications.

In an authentication system, cryptography is used to guarantee the authenticity of the message to the receiver. Not only must a meddler be prevented from injecting totally new, authentic looking messages into a channel, but he must be prevented from creating apparently authentic messages by combining, or merely repeating, old messages which he has copied in the past. A cryptographic system intended to guarantee privacy will not, in general, prevent this latter form of mischief.

To guarantee the authenticity of a message, information is added which is a function not only of the message and a secret key, but of the date and time as well; for example, by attaching the date and time to each message and encrypting the entire sequence. This assures that only someone who possesses the key can generate a message which, when decrypted, will contain the proper date and time. Care must be taken, however, to use a system in which small changes in the ciphertext result in large changes in the deciphered plaintext. This intentional error propagation ensures that if the deliberate injection of noise on the channel changes a message such as "erase file 7" into a different message such as "erase file 8," it will also corrupt the authentication information. The message will then be rejected as inauthentic.

The first step in assessing the adequacy of cryptographic systems is to classify the threats to which they are to be subjected. The following threats may occur to cryptographic systems employed for either privacy or authentication.

A *ciphertext only attack* is a cryptanalytic attack in which the cryptanalyst possesses only ciphertext.

A *known plaintext attack* is a cryptanalytic attack in which the cryptanalyst possesses a substantial quantity of corresponding plaintext and ciphertext.

A *chosen plaintext attack* is a cryptanalytic attack in which the cryptanalyst can submit an unlimited number of plaintext messages of his own choosing and examine the resulting cryptograms.

In all cases it is assumed that the opponent knows the general system $\{S_K\}$ in use since this information can be obtained by studying a cryptographic device. While many users of cryptography attempt to keep their equipment secret, many commercial applications require not only that the general system be public but that it be standard.

A ciphertext only attack occurs frequently in practice. The cryptanalyst uses only knowledge of the statistical properties of the language in use (e.g., in English, the letter e occurs 13 percent of the time) and knowledge of certain "probable" words (e.g., a letter probably begins "Dear Sir:"). It is the weakest threat to which a system can be subjected, and any system which succumbs to it is considered totally insecure.

A system which is secure against a known plaintext attack frees its users from the need to keep their past messages secret, or to paraphrase them prior to declassification. This is an unreasonable burden to place on the system's users, particularly in commercial situations where product announcements or press releases may be sent in encrypted form for later public disclosure. Similar situations in diplomatic correspondence have led to the cracking of many supposedly secure systems. While a known plaintext attack is not always possible, its occurrence is frequent enough that a system which cannot resist it is not considered secure.

A chosen plaintext attack is difficult to achieve in practice, but can be approximated. For example, submitting a proposal to a competitor may result in his enciphering it for transmission to his headquarters. A cipher which is secure against a chosen plaintext attack thus frees its users from concern over whether their opponents can plant messages in their system.

For the purpose of certifying systems as secure, it is appropriate to consider the more formidable cryptanalytic threats as these not only give more realistic models of the working environment of a cryptographic system, but make the assessment of the system's strength easier. Many systems which are difficult to analyze using a ciphertext only attack can be ruled out immediately under known plaintext or chosen plaintext attacks.

As is clear from these definitions, cryptanalysis is a system identification problem. The known plaintext and chosen plaintext attacks correspond to passive and active system identification problems, respectively. Unlike many subjects in which system identification is considered, such as automatic fault diagnosis, the goal in cryptography is to build systems which are difficult, rather than easy, to identify.

The chosen plaintext attack is often called an IFF attack, terminology which descends from its origin in the development of cryptographic "identification friend or foe" systems after World War II. An IFF system enables military radars to distinguish between friendly and enemy planes automatically. The radar sends a time-varying challenge to the airplane which receives the challenge, encrypts it under the appropriate key, and sends it back to the radar. By comparing this response with a correctly encrypted version of the challenge, the radar can recognize a friendly aircraft. While the aircraft are over enemy territory, enemy cryptanalysts can send challenges and examine the encrypted responses in an attempt to determine the authentication key in use, thus mounting a chosen plaintext attack on the system. In practice, this threat is countered by restricting the form of the challenges, which need not be unpredictable, but only nonrepeating.

There are other threats to authentication systems which cannot be treated by conventional cryptography, and which require recourse to the new ideas and techniques introduced in this paper. The *threat of compromise of the receiver's authentication data* is motivated by the situation in multiuser networks where the receiver is often the system itself. The receiver's password tables and other authentication data are then more vulnerable to theft than those of the transmitter (an individual user). As shown later, some techniques for protecting against this threat also protect against the *threat of dispute*. That is, a message may be sent but later repudiated by either the transmitter or the receiver. Or, it may be alleged by either party that a message was sent when in fact none was. Unforgeable digital signatures and receipts are needed. For example, a dishonest stockbroker might try to cover up unauthorized buying and selling for personal gain by forging orders from clients, or a client might disclaim an order actually authorized by him but which he later sees will cause a loss. We will introduce concepts which allow the receiver to verify the authenticity of a message, but prevent him from generating apparently authentic messages, thereby protecting against both the threat of compromise of the receiver's authentication data and the threat of dispute.

III. Public Key Cryptography

As shown in Fig. 1, cryptography has been a derivative security measure. Once a secure channel exists along which keys can be transmitted, the security can be extended to other channels of higher bandwidth or smaller delay by encrypting the messages sent on them. The effect has been to limit the use of cryptography to communications among people who have made prior preparation for cryptographic security.

In order to develop large, secure, telecommunications systems, this must be changed. A large number of users n results in an even larger number, $(n^2 - n)/2$ potential pairs who may wish to communicate privately from all others. It is unrealistic to assume either that a pair of users with no prior acquaintance will be able to wait for a key to be sent by some secure physical means, or that keys for all $(n^2 - n)/2$ pairs can be arranged in advance. In another paper [5], the authors have considered a conservative approach requiring no new development in cryptography itself, but this involves diminished security, inconvenience, and restriction of the network to a starlike configuration with respect to initial connection protocol.

We propose that it is possible to develop systems of the type shown in Fig. 2, in which two parties communicating solely over a public channel and using only publicly known techniques can create a secure connection. We examine two approaches to this problem, called public key cryptosys-

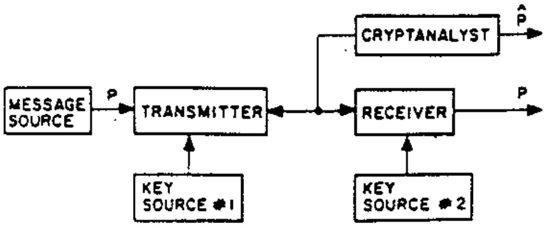


Fig. 2. Flow of information in public key system.

tems and public key distribution systems, respectively. The first are more powerful, lending themselves to the solution of the authentication problems treated in the next section, while the second are much closer to realization.

A *public key cryptosystem* is a pair of families $\{E_K\}_{K \in \{K\}}$ and $\{D_K\}_{K \in \{K\}}$ of algorithms representing invertible transformations,

$$E_K : \{M\} \to \{M\} \tag{2}$$

$$D_K : \{M\} \to \{M\} \tag{3}$$

on a finite message space $\{M\}$, such that

1) for every $K \in \{K\}$, E_K is the inverse of D_K,
2) for every $K \in \{K\}$ and $M \in \{M\}$, the algorithms E_K and D_K are easy to compute,
3) for almost every $K \in \{K\}$, each easily computed algorithm equivalent to D_K is computationally infeasible to derive from E_K,
4) for every $K \in \{K\}$, it is feasible to compute inverse pairs E_K and D_K from K.

Because of the third property, a user's enciphering key E_K can be made public without compromising the security of his secret deciphering key D_K. The cryptographic system is therefore split into two parts, a family of enciphering transformations and a family of deciphering transformations in such a way that, given a member of one family, it is infeasible to find the corresponding member of the other.

The fourth property guarantees that there is a feasible way of computing corresponding pairs of inverse transformations when no constraint is placed on what either the enciphering or deciphering transformation is to be. In practice, the cryptoequipment must contain a true random number generator (e.g., a noisy diode) for generating K, together with an algorithm for generating the $E_K - D_K$ pair from its outputs.

Given a system of this kind, the problem of key distribution is vastly simplified. Each user generates a pair of inverse transformations, E and D, at his terminal. The deciphering transformation D must be kept secret, but need never be communicated on any channel. The enciphering key E can be made public by placing it in a public directory along with the user's name and address. Anyone can then encrypt messages and send them to the user, but no one else can decipher messages intended for him. Public key cryptosystems can thus be regarded as *multiple access ciphers.*

It is crucial that the public file of enciphering keys be protected from unauthorized modification. This task is made easier by the public nature of the file. Read protection is unnecessary and, since the file is modified infrequently, elaborate write protection mechanisms can be economically employed.

A suggestive, although unfortunately useless, example of a public key cryptosystem is to encipher the plaintext, represented as a binary n-vector m, by multiplying it by an invertible binary $n \times n$ matrix E. The cryptogram thus equals Em. Letting $D = E^{-1}$ we have $m = Dc$. Thus, both enciphering and deciphering require about n^2 operations. Calculation of D from E, however, involves a matrix inversion which is a harder problem. And it is at least conceptually simpler to obtain an arbitrary pair of inverse matrices than it is to invert a given matrix. Start with the identity matrix I and do elementary row and column operations to obtain an arbitrary invertible matrix E. Then starting with I do the inverses of these same elementary operations in reverse order to obtain $D = E^{-1}$. The sequence of elementary operations could be easily determined from a random bit string.

Unfortunately, matrix inversion takes only about n^3 operations. The ratio of "cryptanalytic" time (i.e., computing D from E) to enciphering or deciphering time is thus at most n, and enormous block sizes would be required to obtain ratios of 10^6 or greater. Also, it does not appear that knowledge of the elementary operations used to obtain E from I greatly reduces the time for computing D. And, since there is no round-off error in binary arithmetic, numerical stability is unimportant in the matrix inversion. In spite of its lack of practical utility, this matrix example is still useful for clarifying the relationships necessary in a public key cryptosystem.

A more practical approach to finding a pair of easily computed inverse algorithms E and D; such that D is hard to infer from E, makes use of the difficulty of analyzing programs in low level languages. Anyone who has tried to determine what operation is accomplished by someone else's machine language program knows that E itself (i.e., what E does) can be hard to infer from an algorithm for E. If the program were to be made purposefully confusing through addition of unneeded variables and statements, then determining an inverse algorithm could be made very difficult. Of course, E must be complicated enough to prevent its identification from input–output pairs.

Essentially what is required is a one-way compiler: one which takes an easily understood program written in a high level language and translates it into an incomprehensible program in some machine language. The compiler is one-way because it must be feasible to do the compilation, but infeasible to reverse the process. Since efficiency in size of program and run time are not crucial in this application, such compilers may be possible if the structure of the machine language can be optimized to assist in the confusion.

Merkle [1] has independently studied the problem of distributing keys over an insecure channel. His approach is different from that of the public key cryptosystems suggested above, and will be termed a *public key distribution system.* The goal is for two users, A and B, to securely exchange a key over an insecure channel. This key is then used by both users in a normal cryptosystem for both enciphering and deciphering. Merkle has a solution whose cryptanalytic cost grows as n^2 where n is the cost to the legitimate users. Unfortunately the cost to the legitimate users of the system is as much in transmission time as in computation, because Merkle's protocol requires n

potential keys to be transmitted before one key can be decided on. Merkle notes that this high transmission overhead prevents the system from being very useful in practice. If a one megabit limit is placed on the setup protocol's overhead, his technique can achieve cost ratios of approximately 10 000 to 1, which are too small for most applications. If inexpensive, high bandwidth data links become available, ratios of a million to one or greater could be achieved and the system would be of substantial practical value.

We now suggest a new public key distribution system which has several advantages. First, it requires only one "key" to be exchanged. Second, the cryptanalytic effort appears to grow exponentially in the effort of the legitimate users. And, third, its use can be tied to a public file of user information which serves to authenticate user A to user B and vice versa. By making the public file essentially a read only memory, one personal appearance allows a user to authenticate his identity many times to many users. Merkle's technique requires A and B to verify each other's identities through other means.

The new technique makes use of the apparent difficulty of computing logarithms over a finite field $GF(q)$ with a prime number q of elements. Let

$$Y = \alpha^X \bmod q, \qquad \text{for } 1 \le X \le q - 1, \tag{4}$$

where α is a fixed primitive element of $GF(q)$, then X is referred to as the logarithm of Y to the base α, mod q:

$$X = \log_\alpha Y \bmod q, \qquad \text{for } 1 \le Y \le q - 1. \tag{5}$$

Calculation of Y from X is easy, taking at most $2 \times \log_2 q$ multiplications [6, pp. 398–422]. For example, for $X = 18$,

$$Y = \alpha^{18} = (((\alpha^2)^2)^2)^2 \times \alpha^2. \tag{6}$$

Computing X from Y, on the other hand can be much more difficult and, for certain carefully chosen values of q, requires on the order of $q^{1/2}$ operations, using the best known algorithm [7, pp. 9, 575–576], [8].

The security of our technique depends crucially on the difficulty of computing logarithms mod q, and if an algorithm whose complexity grew as $\log_2 q$ were to be found, our system would be broken. While the simplicity of the problem statement might allow such simple algorithms, it might instead allow a proof of the problem's difficulty. For now we assume that the best known algorithm for computing logs mod q is in fact close to optimal and hence that $q^{1/2}$ is a good measure of the problem's complexity, for a properly chosen q.

Each user generates an independent random number X_i chosen uniformly from the set of integers $\{1, 2, \cdots, q - 1\}$. Each keeps X_i secret, but places

$$Y_i = \alpha^{X_i} \bmod q \tag{7}$$

in a public file with his name and address. When users i and j wish to communicate privately, they use

$$K_{ij} = \alpha^{X_i X_j} \bmod q \tag{8}$$

as their key. User i obtains K_{ij} by obtaining Y_j from the public file and letting

$$K_{ij} = Y_j^{X_i} \bmod q \tag{9}$$

$$= (\alpha^{X_j})^{X_i} \bmod q \tag{10}$$

$$= \alpha^{X_j X_i} = \alpha^{X_i X_j} \bmod q. \tag{11}$$

User j obtains K_{ij} in the similar fashion

$$K_{ij} = Y_i^{X_j} \bmod q. \tag{12}$$

Another user must compute K_{ij} from Y_i and Y_j, for example, by computing

$$K_{ij} = Y_i^{(\log_\alpha Y_j)} \bmod q. \tag{13}$$

We thus see that if logs mod q are easily computed the system can be broken. While we do not currently have a proof of the converse (i.e., that the system is secure if logs mod q are difficult to compute), neither do we see any way to compute K_{ij} from Y_i and Y_j without first obtaining either X_i or X_j.

If q is a prime slightly less than 2^b, then all quantities are representable as b bit numbers. Exponentiation then takes at most $2b$ multiplications mod q, while by hypothesis taking logs requires $q^{1/2} = 2^{b/2}$ operations. The cryptanalytic effort therefore grows exponentially relative to legitimate efforts. If $b = 200$, then at most 400 multiplications are required to compute Y_i from X_i, or K_{ij} from Y_i and X_j, yet taking logs mod q requires 2^{100} or approximately 10^{30} operations.

IV. One-Way Authentication

The problem of authentication is perhaps an even more serious barrier to the universal adoption of telecommunications for business transactions than the problem of key distribution. Authentication is at the heart of any system involving contracts and billing. Without it, business cannot function. Current electronic authentication systems cannot meet the need for a purely digital, unforgeable, message dependent signature. They provide protection against third party forgeries, but do not protect against disputes between transmitter and receiver.

In order to develop a system capable of replacing the current written contract with some purely electronic form of communication, we must discover a digital phenomenon with the same properties as a written signature. It must be easy for anyone to recognize the signature as authentic, but impossible for anyone other than the legitimate signer to produce it. We will call any such technique *one-way authentication*. Since any digital signal can be copied precisely, a true digital signature must be recognizable without being known.

Consider the "login" problem in a multiuser computer system. When setting up his account, the user chooses a password which is entered into the system's password directory. Each time he logs in, the user is again asked to provide his password. By keeping this password secret from all other users, forged logins are prevented. This, however, makes it vital to preserve the security of the password directory since the information it contains would allow perfect impersonation of any user. The problem is further compounded if system operators have legitimate reasons for accessing the directory. Allowing such legitimate accesses, but preventing all others, is next to impossible.

This leads to the apparently impossible requirement for a new login procedure capable of judging the authenticity of passwords without actually knowing them. While appearing to be a logical impossibility, this proposal is easily satisfied. When the user first enters his password PW, the computer automatically and transparently computes a function $f(PW)$ and stores this, not PW, in the password directory. At each successive login, the computer calculates $f(X)$, where X is the proffered password, and compares $f(X)$ with the stored value $f(PW)$. If and only if they are equal, the user is accepted as being authentic. Since the function f must be calculated once per login, its computation time must be small. A million instructions (costing approximately $0.10 at bicentennial prices) seems to be a reasonable limit on this computation. If we could ensure, however, that calculation of f^{-1} required 10^{30} or more instructions, someone who had subverted the system to obtain the password directory could not in practice obtain PW from $f(PW)$, and could thus not perform an unauthorized login. Note that $f(PW)$ is not accepted as a password by the login program since it will automatically compute $f(f(PW))$ which will not match the entry $f(PW)$ in the password directory.

We assume that the function f is public information, so that it is not ignorance of f which makes calculation of f^{-1} difficult. Such functions are called one-way functions and were first employed for use in login procedures by R. M. Needham [9, p. 91]. They are also discussed in two recent papers [10], [11] which suggest interesting approaches to the design of one-way functions.

More precisely, a function f is a *one-way function* if, for any argument x in the domain of f, it is easy to compute the corresponding value $f(x)$, yet, for almost all y in the range of f, it is computationally infeasible to solve the equation $y = f(x)$ for any suitable argument x.

It is important to note that we are defining a function which is not invertible from a computational point of view, but whose noninvertibility is entirely different from that normally encountered in mathematics. A function f is normally called "noninvertible" when the inverse of a point y is not unique, (i.e., there exist distinct points x_1 and x_2 such that $f(x_1) = y = f(x_2)$). We emphasize that this is not the sort of inversion difficulty that is required. Rather, it must be overwhelmingly difficult, given a value y and knowledge of f, to calculate any x whatsoever with the property that $f(x) = y$. Indeed, if f is noninvertible in the usual sense, it may make the task of finding an inverse image easier. In the extreme, if $f(x) \equiv y_0$ for all x in the domain, then the range of f is $\{y_0\}$, and we can take any x as $f^{-1}(y_0)$. It is therefore necessary that f not be too degenerate. A small degree of degeneracy is tolerable and, as discussed later, is probably present in the most promising class of one-way functions.

Polynomials offer an elementary example of one-way functions. It is much harder to find a root x_0 of the polynomial equation $p(x) = y$ than it is to evaluate the polynomial $p(x)$ at $x = x_0$. Purdy [11] has suggested the use of sparse polynomials of very high degree over finite fields, which appear to have very high ratios of solution to evaluation time. The theoretical basis for one-way functions is discussed at greater length in Section VI. And, as shown in Section V, one-way functions are easy to devise in practice.

The one-way function login protocol solves only some of the problems arising in a multiuser system. It protects against compromise of the system's authentication data when it is not in use, but still requires the user to send the true password to the system. Protection against eavesdropping must be provided by additional encryption, and protection against the threat of dispute is absent altogether.

A public key cryptosystem can be used to produce a true one-way authentication system as follows. If user A wishes to send a message M to user B, he "deciphers" it in his secret deciphering key and sends $D_A(M)$. When user B receives it, he can read it, and be assured of its authenticity by "enciphering" it with user A's public enciphering key E_A. B also saves $D_A(M)$ as proof that the message came from A. Anyone can check this claim by operating on $D_A(M)$ with the publicly known operation E_A to recover M. Since only A could have generated a message with this property, the solution to the one-way authentication problem would follow immediately from the development of public key cryptosystems.

One-way message authentication has a partial solution suggested to the authors by Leslie Lamport of Massachusetts Computer Associates. This technique employs a one-way function f mapping k-dimensional binary space into itself for k on the order of 100. If the transmitter wishes to send an N bit message he generates $2N$, randomly chosen, k-dimensional binary vectors $x_1,X_1,x_2,X_2,\cdots,x_N,X_N$ which he keeps secret. The receiver is given the corresponding images under f, namely $y_1,Y_1,y_2,Y_2,\cdots,y_N,Y_N$. Later, when the message $\boldsymbol{m} = (m_1,m_2,\cdots,m_N)$ is to be sent, the transmitter sends x_1 or X_1 depending on whether $m_1 = 0$ or 1. He sends x_2 or X_2 depending on whether $m_2 = 0$ or 1, etc. The receiver operates with f on the first received block and sees whether it yields y_1 or Y_1 as its image and thus learns whether it was x_1 or X_1, and whether $m_1 = 0$ or 1. In a similar manner the receiver is able to determine $m_2,m_3,\cdots,m_N$. But the receiver is incapable of forging a change in even one bit of $\boldsymbol{m}$.

This is only a partial solution because of the approximately 100-fold data expansion required. There is, however, a modification which eliminates the expansion problem when N is roughly a megabit or more. Let g be a one-way mapping from binary N-space to binary n-space where n is approximately 50. Take the N bit message $\boldsymbol{m}$

and operate on it with g to obtain the n bit vector m'. Then use the previous scheme to send m'. If $N = 10^6$, $n = 50$, and $k = 100$, this adds $kn = 5000$ authentication bits to the message. It thus entails only a 5 percent data expansion during transmission (or 15 percent if the initial exchange of $y_1, Y_1, \cdots, y_N, Y_N$ is included). Even though there are a large number of other messages (2^{N-n} on the average) with the same authentication sequence, the one-wayness of g makes them computationally infeasible to find and thus to forge. Actually g must be somewhat stronger than a normal one-way function, since an opponent has not only m' but also one of its inverse images m. It must be hard even given m to find a different inverse image of m'. Finding such functions appears to offer little trouble (see Section V).

There is another partial solution to the one-way user authentication problem. The user generates a password X which he keeps secret. He gives the system $f^T(X)$, where f is a one-way function. At time t the appropriate authenticator is $f^{T-t}(X)$, which can be checked by the system by applying $f^t(X)$. Because of the one-wayness of f, past responses are of no value in forging a new response. The problem with this solution is that it can require a fair amount of computation for legitimate login (although many orders of magnitude less than for forgery). If for example t is incremented every second and the system must work for one month on each password then $T = 2.6$ million. Both the user and the system must then iterate f an average of 1.3 million times per login. While not insurmountable, this problem obviously limits use of the technique. The problem could be overcome if a simple method for calculating $f^{(2\uparrow n)}$, for $n = 1,2, \cdots$ could be found, much as $X^8 = ((X^2)^2)^2$. For then binary decompositions of $T - t$ and t would allow rapid computation of f^{T-t} and f^t. It may be, however, that rapid computation of f^n precludes f from being one-way.

V. Problem Interrelations and Trap Doors

In this section, we will show that some of the cryptographic problems presented thus far can be reduced to others, thereby defining a loose ordering according to difficulty. We also introduce the more difficult problem of trap doors.

In Section II we showed that a cryptographic system intended for privacy can also be used to provide authentication against third party forgeries. Such a system can be used to create other cryptographic objects, as well.

A cryptosystem which is secure against a known plaintext attack can be used to produce a one-way function.

As indicated in Fig. 3, take the cryptosystem $\{S_K:\{P\} \rightarrow \{C\}\}_{K \in \{K\}}$ which is secure against a known plaintext attack, fix $P = P_0$ and consider the map

$$f:\{K\} \rightarrow \{C\} \tag{14}$$

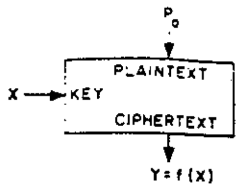


Fig. 3. Secure cryptosystem used as one-way function.

defined by

$$f(X) = S_X(P_0). \tag{15}$$

This function is one-way because solving for X given $f(X)$ is equivalent to the cryptanalytic problem of finding the key from a single known plaintext–cryptogram pair. Public knowledge of f is now equivalent to public knowledge of $\{S_K\}$ and P_0.

While the converse of this result is not necessarily true, it is possible for a function originally found in the search for one-way functions to yield a good cryptosystem. This actually happened with the discrete exponential function discussed in Section III [8].

One-way functions are basic to both block ciphers and key generators. A key generator is a pseudorandom bit generator whose output, the keystream, is added modulo 2 to a message represented in binary form, in imitation of a one-time pad. The key is used as a "seed" which determines the pseudorandom keystream sequence. A known plaintext attack thus reduces to the problem of determining the key from the keystream. For the system to be secure, computation of the key from the keystream must be computationally infeasible. While, for the system to be usable, calculation of the keystream from the key must be computationally simple. Thus a good key generator is, almost by definition, a one-way function.

Use of either type of cryptosystem as a one way function suffers from a minor problem. As noted earlier, if the function f is not uniquely invertible, it is not necessary (or possible) to find the actual value of X used. Rather any X with the same image will suffice. And, while each mapping S_K in a cryptosystem must be bijective, there is no such restriction on the function f from key to cryptogram defined above. Indeed, guaranteeing that a cryptosystem has this property appears quite difficult. In a good cryptosystem the mapping f can be expected to have the characteristics of a randomly chosen mapping (i.e., $f(X_i)$ is chosen uniformly from all possible Y, and successive choices are independent). In this case, if X is chosen uniformly and there are an equal number of keys and messages (X and Y), then the probability that the resultant Y has $k + 1$ inverses is approximately $e^{-1}/k!$ for $k = 0,1,2,3,\cdots$. This is a Poisson distribution with mean $\lambda = 1$, shifted by 1 unit. The expected number of inverses is thus only 2. While it is possible for f to be more degenerate, a good cryptosystem will not be too degenerate since then the key is not being well used. In the worst case, if $f(X) \equiv Y_0$ for some Y_0, we have $S_K(P_0) \equiv C_0$, and encipherment of P_0 would not depend on the key at all!

While we are usually interested in functions whose domain and range are of comparable size, there are exceptions. In the previous section we required a one-way function mapping long strings onto much shorter ones. By using a block cipher whose key length is larger than the blocksize, such functions can be obtained using the above technique.

Evans *et al.* [10] have a different approach to the problem of constructing a one-way function from a block cipher. Rather than selecting a fixed P_0 as the input, they use the function

$$f(X) = S_X(X). \qquad (16)$$

This is an attractive approach because equations of this form are generally difficult to solve, even when the family S is comparatively simple. This added complexity, however, destroys the equivalence between the security of the system S under a known plaintext attack and the one-wayness of f.

Another relationship has already been shown in Section IV.

A public key cryptosystem can be used to generate a one-way authentication system.

The converse does not appear to hold, making the construction of a public key cryptosystem a strictly more difficult problem than one-way authentication. Similarly, a public key cryptosystem can be used as a public key distribution system, but not conversely.

Since in a public key cryptosystem the general system in which E and D are used must be public, specifying E specifies a complete algorithm for transforming input messages into output cryptograms. As such a public key system is really a set of *trap-door one-way functions*. These are functions which are not really one-way in that simply computed inverses exist. But given an algorithm for the forward function it is computationally infeasible to find a simply computed inverse. Only through knowledge of certain *trap-door information* (e.g., the random bit string which produced the E-D pair) can one easily find the easily computed inverse.

Trap doors have already been seen in the previous paragraph in the form of *trap-door one-way functions*, but other variations exist. A *trap-door cipher* is one which strongly resists cryptanalysis by anyone not in possession of *trap-door information* used in the design of the cipher. This allows the designer to break the system after he has sold it to a client and yet falsely to maintain his reputation as a builder of secure systems. It is important to note that it is not greater cleverness or knowledge of cryptography which allows the designer to do what others cannot. If he were to lose the trap-door information he would be no better off than anyone else. The situation is precisely analogous to a combination lock. Anyone who knows the combination can do in seconds what even a skilled locksmith would require hours to accomplish. And yet, if he forgets the combination, he has no advantage.

A trap-door cryptosystem can be used to produce a public key distribution system.

For A and B to establish a common private key, A chooses a key at random and sends an arbitrary plaintext–cryptogram pair to B. B, who made the trap-door cipher public, but kept the trap-door information secret, uses the plaintext-cryptogram pair to solve for the key. A and B now have a key in common.

There is currently little evidence for the existence of trap-door ciphers. However they are a distinct possibility and should be remembered when accepting a cryptosystem from a possible opponent [12].

By definition, we will require that a trap-door problem be one in which it is computationally feasible to devise the trap door. This leaves room for yet a third type of entity for which we shall use the prefix "quasi." For example a *quasi one-way function* is not one-way in that an easily computed inverse exists. However, it is computationally infeasible even for the designer, to find the easily computed inverse. Therefore a quasi one-way function can be used in place of a one-way function with essentially no loss in security.

Losing the trap-door information to a trap-door one-way function makes it into a quasi one-way function, but there may also be one-way functions not obtainable in this manner.

It is entirely a matter of definition that quasi one-way functions are excluded from the class of one-way functions. One could instead talk of one-way functions in the wide sense or in the strict sense.

Similarly, a quasi secure cipher is a cipher which will successfully resist cryptanalysis, even by its designer, and yet for which there exists a computationally efficient cryptanalytic algorithm (which is of course computationally infeasible to find). Again, from a practical point of view, there is essentially no difference between a secure cipher and a quasi secure one.

We have already seen that public key cryptosystems imply the existence of trap-door one-way functions. However the converse is not true. For a trap-door one-way function to be usable as a public key cryptosystem, it must be invertible (i.e., have a unique inverse.)

VI. Computational Complexity

Cryptography differs from all other fields of endeavor in the ease with which its requirements may appear to be satisfied. Simple transformations will convert a legible text into an apparently meaningless jumble. The critic, who wishes to claim that meaning might yet be recovered by cryptanalysis, is then faced with an arduous demonstration if he is to prove his point of view correct. Experience has shown, however, that few systems can resist the concerted attack of skillful cryptanalysts, and many supposedly secure systems have subsequently been broken.

In consequence of this, judging the worth of new systems has always been a central concern of cryptographers. During the sixteenth and seventeenth centuries, mathematical arguments were often invoked to argue the strength of cryptographic methods, usually relying on counting methods which showed the astronomical number of possible keys. Though the problem is far too difficult to be laid to rest by such simple methods, even the noted algebraist Cardano fell into this trap [2, p. 145]. As systems whose strength had been so argued were repeatedly broken, the notion of giving mathematical proofs for the security of systems fell into disrepute and was replaced by certification via crypanalytic assault.

During this century, however, the pendulum has begun to swing back in the other direction. In a paper intimately connected with the birth of information theory, Shannon [3] showed that the one time pad system, which had been in use since the late twenties offered "perfect secrecy" (a form of unconditional security). The provably secure systems investigated by Shannon rely on the use of either a key whose length grows linearly with the length of the message or on perfect source coding and are therefore too unwieldy for most purposes. We note that neither public key cryptosystems nor one-way authentication systems can be unconditionally secure because the public information always determines the secret information uniquely among the members of a finite set. With unlimited computation, the problem could therefore be solved by a straightforward search.

The past decade has seen the rise of two closely related disciplines devoted to the study of the costs of computation: computational complexity theory and the analysis of algorithms. The former has classified known problems in computing into broad classes by difficulty, while the latter has concentrated on finding better algorithms and studying the resources they consume. After a brief digression into complexity theory, we will examine its application to cryptography, particularly the analysis of one-way functions.

A function is said to belong to the complexity class ***P*** (for polynomial) if it can be computed by a deterministic Turing Machine in a time which is bounded above by some polynomial function of the length of its input. One might think of this as the class of easily computed functions, but it is more accurate to say that a function not in this class must be hard to compute for at least some inputs. There are problems which are known not to be in the class ***P*** [13, pp. 405–425].

There are many problems which arise in engineering which cannot be solved in polynomial time by any known techniques, unless they are run on a computer with an unlimited degree of parallelism. These problems may or may not belong to the class ***P***, but belong to the class ***NP*** (for nondeterministic, polynomial) of problems solvable in polynomial time on a "nondeterministic" computer (i.e., one with an unlimited degree of parallelism). Clearly the class ***NP*** includes the class ***P***, and one of the great open questions in complexity theory is whether the class ***NP*** is strictly larger.

Among the problems known to be solvable in ***NP*** time, but not known to be solvable in ***P*** time, are versions of the traveling salesman problem, the satisfiability problem for propositional calculus, the knapsack problem, the graph coloring problem, and many scheduling and minimization problems [13, pp. 363–404], [14]. We see that it is not lack of interest or effort which has prevented people from finding solutions in ***P*** time for these problems. It is thus strongly believed that at least one of these problems must not be in the class ***P***, and that therefore the class ***NP*** is strictly larger.

Karp has identified a subclass of the ***NP*** problems, called ***NP*** complete, with the property that if any one of them is in ***P***, then all ***NP*** problems are in ***P***. Karp lists 21 problems which are ***NP*** complete, including all of the problems mentioned above [14].

While the ***NP*** complete problems show promise for cryptographic use, current understanding of their difficulty includes only worst case analysis. For cryptographic purposes, typical computational costs must be considered. If, however, we replace worst case computation time with average or typical computation time as our complexity measure, the current proofs of the equivalences among the ***NP*** complete problems are no longer valid. This suggests several interesting topics for research. The ensemble and typicality concepts familiar to information theorists have an obvious role to play.

We can now identify the position of the general cryptanalytic problem among all computational problems.

The cryptanalytic difficulty of a system whose encryption and decryption operations can be done in ***P*** *time cannot be greater than* ***NP***.

To see this, observe that any cryptanalytic problem can be solved by finding a key, inverse image, etc., chosen from a finite set. Choose the key nondeterministically and verify in ***P*** time that it is the correct one. If there are M possible keys to choose from, an M-fold parallelism must be employed. For example in a known plaintext attack, the plaintext is encrypted simultaneously under each of the keys and compared with the cryptogram. Since, by assumption, encryption takes only ***P*** time, the cryptanalysis takes only ***NP*** time.

We also observe that the general cryptanalytic problem is ***NP*** complete. This follows from the breadth of our definition of cryptographic problems. A one-way function with an ***NP*** complete inverse will be discussed next.

Cryptography can draw directly from the theory of ***NP*** complexity by examining the way in which ***NP*** complete problems can be adapted to cryptographic use. In particular, there is an ***NP*** complete problem known as the knapsack problem which lends itself readily to the construction of a one-way function.

Let $y = f(x) = \mathbf{a} \cdot \mathbf{x}$ where $\mathbf{a}$ is a known vector of n intergers $(a_1, a_2, \cdots, a_n)$ and $\mathbf{x}$ is a binary n-vector. Calculation of y is simple, involving a sum of at most n integers. The problem of inverting f is known as the knapsack problem and requires finding a subset of the $\{a_i\}$ which sum to y.

Exhaustive search of all 2^n subsets grows exponentially and is computationally infeasible for n greater than 100 or so. Care must be exercised, however, in selecting the parameters of the problem to ensure that shortcuts are not possible. For example if $n = 100$ and each a_i is 32 bits long, y is at most 39 bits long, and f is highly degenerate; requiring on the average only 2^{38} tries to find a solution. Somewhat more trivially, if $a_i = 2^{i-1}$ then inverting f is equivalent to finding the binary decomposition of y.

This example demonstrates both the great promise and the considerable shortcomings of contemporary complexity theory. The theory only tells us that the knapsack problem is probably difficult in the worst case. There is no indication of its difficulty for any particular array. It appears, however, that choosing the $\{a_i\}$ uniformly from $\{0,1,2,\cdots,2^{n-1}\}$ results in a hard problem with probability one as $n \rightarrow \infty$.

Another potential one-way function, of interest in the analysis of algorithms, is exponentiation mod q, which was suggested to the authors by Prof. John Gill of Stanford University. The one-wayness of this functions has already been discussed in Section III.

VII. Historical Perspective

While at first the public key systems and one-way authentication systems suggested in this paper appear to be unportended by past cryptographic developments, it is possible to view them as the natural outgrowth of trends in cryptography stretching back hundreds of years.

Secrecy is at the heart of cryptography. In early cryptography, however, there was a confusion about what was to be kept secret. Cryptosystems such as the Caesar cipher (in which each letter is replaced by the one three places further on, so A is carried to D, B to E, etc.) depended for their security on keeping the entire encryption process secret. After the invention of the telegraph [2, p. 191], the distinction between a general system and a specific key allowed the general system to be compromised, for example by theft of a cryptographic device, without compromising future messages enciphered in new keys. This principle was codified by Kerchoffs [2, p. 235] who wrote in 1881 that the compromise of a cryptographic system should cause no inconvenience to the correspondents. About 1960, cryptosystems were put into service which were deemed strong enough to resist a known plaintext cryptanalytic attack, thereby eliminating the burden of keeping old messages secret. Each of these developments decreased the portion of the system which had to be protected from public knowledge, eliminating such tedious expedients as paraphrasing diplomatic dispatches before they were presented. Public key systems are a natural continuation of this trend toward decreasing secrecy.

Prior to this century, cryptographic systems were limited to calculations which could be carried out by hand or with simple slide-rule-like devices. The period immediately after World War I saw the beginning of a revolutionary trend which is now coming to fruition. Special purpose machines were developed for enciphering. Until the development of general purpose digital hardware, however, cryptography was limited to operations which could be performed with simple electromechanical systems. The development of digital computers has freed it from the limitations of computing with gears and has allowed the search for better encryption methods according to purely cryptographic criteria.

The failure of numerous attempts to demonstrate the soundness of cryptographic systems by mathematical proof led to the paradigm of certification by cryptanalytic attack set down by Kerchoffs [2, p. 234] in the last century. Although some general rules have been developed, which aid the designer in avoiding obvious weaknesses, the ultimate test is an assault on the system by skilled cryptanalysts under the most favorable conditions (e.g., a chosen plaintext attack). The development of computers has led for the first time to a mathematical theory of algorithms which can begin to approach the difficult problem of estimating the computational difficulty of breaking a cryptographic system. The position of mathematical proof may thus come full circle and be reestablished as the best method of certification.

The last characteristic which we note in the history of cryptography is the division between amateur and professional cryptographers. Skill in production cryptanalysis has always been heavily on the side of the professionals, but innovation, particularly in the design of new types of cryptographic systems, has come primarily from the amateurs. Thomas Jefferson, a cryptographic amateur, invented a system which was still in use in World War II [2, pp. 192–195], while the most noted cryptographic system of the twentieth century, the rotor machine, was invented simultaneously by four separate people, all amateurs [2, pp. 415, 420, 422–424]. We hope this will inspire others to work in this fascinating area in which participation has been discouraged in the recent past by a nearly total government monopoly.

References

[1] R. Merkle, "Secure communication over an insecure channel," submitted to *Communications of the ACM*.

[2] D. Kahn, *The Codebreakers, The Story of Secret Writing*. New York: Macmillan, 1967.

[3] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

[4] M. E. Hellman, "An extension of the Shannon theory approach to cryptography," submitted to *IEEE Trans. Inform. Theory*, Sept. 1975.

[5] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques," presented at National Computer Conference, New York, June 7–10, 1976.

[6] D. Knuth, *The Art of Computer Programming, Vol. 2, Semi-Numerical Algorithms*. Reading, MA.: Addison-Wesley, 1969.

[7] ——, *The Art of Computer Programming, Vol. 3, Sorting and Searching*. Reading, MA.: Addison-Wesley, 1973.

[8] S. Pohlig and M. E. Hellman, "An improved algorithm for computing algorithms in $GF(p)$ and its cryptographic significance," submitted to *IEEE Trans. Inform. Theory*.

[9] M. V. Wilkes, *Time-Sharing Computer Systems*. New York: Elsevier, 1972.

[10] A. Evans, Jr., W. Kantrowitz, and E. Weiss, "A user authentication system not requiring secrecy in the computer," *Communications of the ACM*, vol. 17, pp. 437–442, Aug. 1974.

[11] G. B. Purdy, "A high security log-in procedure," *Communications of the ACM*, vol. 17, pp. 442–445, Aug. 1974.

[12] W. Diffie and M. E. Hellman, "Cryptanalysis of the NBS data encryption standard" submitted to *Computer*, May 1976.

[13] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA.: Addison-Wesley, 1974.

[14] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972, pp. 85–104.

4

IN THE MATTER OF THE ARBITRATION BETWEEN

CYLINK CORPORATION AND CARO-KANN CORPORATION, ON THE ONE HAND

AND

RSA DATA SECURITY, INC., ON THE OTHER HAND

[draft] ORDER AND DECISION REGARDING LIABILITY, DISSOLUTION OF PKP AND DECISION ON RSA'S MOTION FOR JUDGMENT AND CYLINK'S CROSS MOTION FOR JUDGMENT

The parties, through their respective counsel, have appeared before the full arbitration panel (the "Panel") at hearings conducted from May 9, 1995 to June 8, 1995 and the Panel has considered the testimony, arguments and evidence regarding liability of the various parties for acts and failures to act under the various agreements connecting the parties. In addition, RSA filed a Motion for Judgment ("Motion for Judgment") on May 18, 1995 and Cylink / Caro-Kann filed an Opposition of Cylink & Caro-Kann to Motion of RSA Data Security, Inc. for Judgment and Cross-motion for Judgment ("Cylink Opposition").

THE PANEL NOW FINDS AND ORDERS as follows:

1. In formulating this Order and Decision, the Panel has reviewed the various arbitration demands of the parties and has reviewed the various testimony and positions taken at the closing arguments which occurred on June 8, 1995. The Panel has determined the questions presented fall essentially into the following categories:

(a) Should Public Key Partners ("PKP") be dissolved?

(b) If there is a decision to dissolve PKP, how should the assets of PKP be distributed and how should the winding up of the partnership affairs of PKP be managed?

(c) Is Cylink entitled to license under the MIT patents and, if so, what should be the extent of any such license? and

(d) Is either party liable to the other for any damages resulting from breach or lack of performance under the various agreements entered into between the parties?

(e) Is either party entitled to collect from the other party any of its fees and costs in connection with the arbitration?

2. <u>Dissolution of PKP</u>: There are three avenues under which PKP may be dissolved.

(a) <u>Dissolution by agreement</u>: First, PKP may be dissolved by agreement of the parties. Although this avenue was explored during an earlier hearing, it was clear that the parties were not in agreement on the terms under which PKP should be dissolved. Therefore, the Panel finds that PKP will not be dissolved by agreement of the parties.

(b) <u>Dissolution as a result of a "Terminating Event"</u>: Second, PKP may be dissolved as a result of a "Terminating Event" as that term is used in the General Partnership Agreement. There is only one Terminating Event which has been put forward by either party--the commission of a Material Breach by a Partner. A review of the General Partnership Agreement shows that the only Material Breach which is put forward by either party is a breach of Article 6, Paragraph 12. Both parties are accused by the other party of breaching Article 6, Paragraph 12. Although the Panel has reviewed the various testimony, evidence and arguments put forward by both parties regarding breach of Article 6, Paragraph 12, the Panel does not find sufficient evidence to hold that either party has breached Article 6, Paragraph 12 or, at least to the extent a party has breached this paragraph, the other party has as well and the Panel would find the parties to be equally at fault for breach. For this reason, the Panel finds that PKP will not be dissolved as a result of a Terminating Event and, in particular, will not be dissolved as a result of a Material Breach of Article 6, Paragraph 12. Therefore, the purchase rights contemplated by Article 9, Paragraph 5(d) are not available to either party. In addition to the above-stated reason for not finding a breach of Article 6, Paragraph 12, the Panel does not find any sufficient evidence of notice by either party to the other party to find compliance with Article 9, Paragraph 4.

(c) <u>Dissolution pursuant to the Panel's equitable powrs:</u> Third, PKP may be dissolved pursuant to the Panel's equitable powers pursuant to Article 12, Paragraph 1. Based on the testimony, evidence and arguments of the parties, it is clear to the Panel that PKP cannot continue as a partnership and, as an matter of equity, should be dissolved. The parties appear to agree that, in any event at the conclusion of this arbitration process, PKP should be dissolved or otherwise wound up. See, e.g., Hearing Transcript, June 8, 1995, pp. 2069-2070:

> ARBITRATOR BUNSOW: . . . What would be the effect of a finding that neither party has committed a material breach? Would it end in a voluntary dissolution?
>
> MR. FLINN: I believe we then proceed down the path of a voluntary consensual dissolution. Neither party wants to be in a partnership with the other, and whatever the law provides for or the contract provides for, the effect of a consensual dissolution is the result.
>
> ARBITRATOR BUNSOW: Do you concur?
>
> MR. BUSSELLE: I know the Panel is going to be shocked, but I absolutely agree.

Therefore, pursuant to the Panel's equitable powers under Article 12, Paragraph 1, the Panel hereby orders that PKP is dissolved effective on the date of this order.

3. <u>Distribution of Assets of PKP</u>: Having determined and ordered that PKP is dissolved pursuant to the Panel's equitable powers, the Panel is now faced with the question of how assets of PKP shall be divided. As has been stated above, to the extent the Panel has generally found any evidence of fault on the part of one party, it has also found evidence of fault on the part of the other party. Therefore, the Panel is convinced that liability, if any, resulting from breach of the various agreements can be dealt with through money damages payable from one party to the other. Liability will be dealt with in greater detail below. However, excepting liabilities and damages resulting therefrom, the Panel is of the opinion that the fairest distribution of assets of PKP will be in accordance with the terms of the General Partnership Agreement for dissolution and liquidation of the partnership where there is not a purchase of the assets of the partnership by one party or the other (see, Article 9, Paragraph 6). These are the terms the parties negotiated for and agreed to prior to formation of this PKP and the Panel is, therefore, of the opinion that these terms represent overall fair terms for liquidation of the assets under the present circumstances. Therefore, the Panel orders distribution of the assets of PKP as follows:

a) Pursuant to Article 9, Paragraph 8 of the General Partnership Agreement, the "Licensed Rights" as that term is used in the General Partnership Agreement shall be distributed to the Partners as provided in the Cylink License Agreement and the RSA License Agreement. RSA will continue to receive 20% of all royalties received by CKC, its Affiliates or assignees (including Cylink) from any sublicenses covered by the Stanford Agreement entered into after the date of this order. These payments shall be made in quarterly payments on the last day of the month following the end of each calendar quarter for any royalties received in the prior calendar quarter and shall be accompanied by a royalty report detailing the basis on which the payment to RSA has been calculated. RSA shall have a reasonable right to an accounting.

b) Other assets of PKP shall be distributed in accordance with Article 5, Paragraph 2 of the General Partnership Agreement. For clarification, debts and obligations of PKP to creditors other than the partners including any costs, fees, and settlements for any outstanding litigation to which PKP is a party shall be paid first. No assets of PKP shall be distributed to the parties pursuant to Article 5, Paragraph 2 until all such litigation is settled or until the parties agree on a distribution formula.

c) In accordance with Article 5, Paragraph 2(d), an agent shall be appointed to collect and distribute any royalties. The parties shall meet and confer regarding appointment of an agreeable agent within ten (10) days from the date of this order and, if they cannot agree, each party shall submit the names and a summary of the qualification of two (2) agents to the Panel within fifteen (15) days from the date of this order. The Panel will then select an agent to act in accordance with Article 5, Paragraph 2(d).

d) Additionally, the Panel has considered the disposition of the agreement entered into between Dr. Schnorr and PKP, since this agreement is an asset of PKP. Since neither Cylink/Caro-Kann nor RSA is awarded the right to purchase partnership assets, it follows that neither has superior rights as against the other to acquire rights to the Schnorr patent. Therefore, the Panel orders as follows:

(I) Immediately, upon reciept of this order, both parties shall notify Dr. Schnorr that PKP has been dissolved; and

(ii) As part of such notification, Dr. Schnorr shall be told that he may, if he desires, terminate his agreement with PKP effective immediately; and

(iii) In the absence of Dr. Schnorr terminating the agreement, the agreement shall be considered an asset of PKP subject to management and disposition by the agent discussed above in Paragraph 3(c).

4. Cylink's right to a license under the MIT Patent: Cylink has argued it is entitled to a license under United States Patent Number 4,405,829 (the "MIT Patent") as a result of the Agreement of Intent, Paragraph 4.3(a)(iv). To quote the language of that paragraph:

> "(iv) To Cylink, an option to sublicense the right to make, use and sell products which would otherwise infringe on the Licensed Rights presently licensed to RSA under the MIT Agreement on terms acceptable to Cylink, which acceptance can not be unreasonably withheld."

This option was to be delivered at closing according to the terms of the Agreement of Intent. At closing a document was executed by both parties (Cylink Ex. 6) which states in pertinent part:

> ". . . as required by Section 4.3(a)(iv) of the Agreement of Intent . . . RSA Data Security, Inc. hereby grants to Cylink Corporation ("CYLINK") an irrevocable option to license the right to make, use and sell products incorporating software provided by RSA practicing methods described in US Patent #4,405,829 by Rivest et al."

The parties are in dispute regarding whether the option provided in Exhibit 6 is the option which RSA was required to deliver under the terms of the Agreement of Intent. The panel has heard testimony, arguments and reviewed evidence regarding this point and the panel finds and orders as follows:

(a) The Agreement of Intent was an integrated agreement at least by virtue of the integration clause included at Article 12, Paragraph 12 of the General Partnership Agreement. The integrated agreement includes all written provisions of the General Partnership Agreement, the Agreement of Intent, Exhibit 6, the Cylink License Agreement and the RSA License Agreement.

(b) The language of Exhibit 6 unambiguously grants to Cylink an option to license the right to make, use and sell products which would otherwise infringe on the Licensed Rights licensed to RSA under the MIT agreement. This language is an option for a patent license, although a limited one. It is limited in the sense that, should Cylink choose to exercise its option, only products made, used or sold by Cylink incorporating software provided to Cylink by RSA would be licensed. Exhibit 6 was signed by an officer of Cylink and, therefore, must have been reasonably acceptable to Cylink. Therefore, the Panel finds that the option to license granted by Exhibit 6 fulfilled the obligations of RSA under Section 4.3(a)(iv) of the Agreement of Intent. To clarify a point, the parties in the RSA Motion for Judgment and in the Cylink Opposition both appear to base arguments somehow around the question of the effect of Exhibit 6 as a "software license". The Panel has found that Exhibit 6 is an option for a patent license and, therefore, the question of whether the grant of an option for a software license is effective as the delivered required under the Agreement of Intent, Paragraph 4.3(a)(iv) is moot. The Panel recognizes the argument raised by Cylink in the Cylink Opposition that to change paragraph 4.3(a)(iv) to require a software license would controvert the intention of the parties. *See, Cylink Opposition, page 6.* However, again, the Panel finds what was granted by RSA was an option for a patent license. Cylink, in the Cylink Opposition, makes the statement that "all parties acknowledged that the software license option was ***not*** the patent sublicense option required by the Agreement of Intent." (emphasis original). *See, Cylink Opposition, page 6.* However, the Panel does not find evidence of such agreement and, even if it did, the Panel is of the opinion that the integrated agreements themselves are clear that an option for a patent license was required and an option for a patent license was granted and any parole evidence, even evidence of later agreement that Exhibit 6 was not a patent license, has no substantive effect.

(c) Because the Panel finds the Agreement of Intent is an integrated agreement and the Panel finds the language of the Agreement of Intent and the language of Exhibit 6 is unambiguous, the Panel is not persuaded to consider the various arguments of Cylink relating to verbal statements that may or may not have been made before, during or after execution of the Agreement of Intent. The Panel, in essence agrees with the position of RSA, that "the agreements signed at the April 6, 1990 closing were a 'final, complete and exclusive' statement of the parties' agreement" (see RSA's Motion for Judgment, pp. 6) at least as to this issue.

Cylink argues in its closing brief (See page 5) that RSA is estopped from refusing to grant to Cylink a patent license because Cylink reasonably relied to its detriment on the undisputed representation that Cylink could always have a patent license. To the extent it is undisputed that RSA made such a representation, in a light most favorable to Cylink, there is a dispute regarding what the terms of such a patent license would be. To now say that Cylink is entitled to a license because it reasonably relied, to its detriment, on statements of RSA that a patent license would be available to Cylink without having clear agreement as to the terms of such license would be to say that a businessman is entitled to purchase a product, sale of which product typically is the subject of significant negotiation, simply because he was told by the owner of the product that it could be purchased without further discussion of price and other terms. This Panel simply does not find it to have been reasonable for Cylink to have relied to its detriment on statements such as "You can always have a patent license" absent evidence that the parties had also reached agreement as to what the terms of such license would be. The Panel does not find any evidence to show such an agreement was reached and, in fact, the evidence is to the contrary.

Cylink also argues in its closing brief (again, see page 5) that RSA was obligated, as a fiduciary in the PKP partnership, to allow PKP to offer a license under the MIT patent to Cylink. The Panel finds that to the extent there was an obligation on the part of RSA to allow PKP to offer a license to Cylink, PKP did offer such a license when it offered the "LEMCOM type" license to Cylink. The terms of the "LEMCOM type" license were apparently unacceptable to Cylink and Cylink did not accept the license offer. The panel finds that RSA was not obligated to offer terms different than the "LEMCOM type" license to Cylink as a result of any fiduciary obligation of RSA. Today, as a result of the dissolution of PKP, RSA no longer has a fiduciary obligation to offer any license, LEMCOM type or otherwise, to Cylink although it may obviously choose to do so.

We will now briefly touch on the question of whether Cylink is entitled to a license under the MIT patent as a result of any commitments PKP may have made to license the MIT patent on a non-discriminatory or similar basis. There has been testimony and evidence that PKP did commit to standards organizations (ANSI) that it would license the MIT patent on a non-discriminatory basis if the invention claimed by the patent were adopted as a standard. Cylink has raised the question of whether the LEMCOM type license is non-discriminatory. The Panel does not find a need to address this question because the offer to the standards committee was subsequently withdrawn and, in any event, the invention claimed by the patent was not adopted as a standard. Therefore, at this point in time, PKP is not obligated as a result of agreements with standards organizations to license Cylink under the MIT patent.

To respond and rule on the May 18, 1995 Motion for Judgment and on Cylink's Cross Motion for Judgment, RSA asked this Panel for judgment on the issue of whether Cylink is entitled to a license under the April 6, 1990 agreements that effected the formation of PKP. In its motion, RSA urged this Panel to order that Cylink is not entitled to any license other than the license offered in Exhibit 6. In its opposition, Cylink took the position that RSA's Motion for Judgment could be granted only if paragraph 4.3(a)(iv) of the Agreement of Intent was written out of the contract. See, Cylink Opposition at page 1. Cylink argues and moves in the Cylink Opposition that it is entitled to judgment that RSA must deliver to Cylink an option to sublicense the MIT patent on terms acceptable to Cylink. For all of the reasons stated above, the Panel disagrees with Cylink's position that the Panel would be writing paragraph 4.3(a)(iv) out of the agreement by granting RSA's Motion for Judgment. Therefore, the Panel hereby grants RSA's Motion for Judgment and denies Cylink's Cross Motion for Judgment.

5. <u>Liabilities of the parties for breach of the agreements</u>: The Panel is now faced with the question of whether either party is liable to the other for any breach of the various provisions of the integrated agreements. Based on closing hearing transcript at pages 2062-2065, the Panel understands the issues regarding

liability which the parties are now asking for a decision on to be those issues for which relief is requested at page 20 of the Cylink Closing Brief and at page 20 of the RSA closing brief .

(a) Liability on the part of RSA: The Issues which the Panel has been requested to decide by Cylink, and the Panel's findings and orders regarding those issues are as follows:

(i) MIT patent license: Cylink has requested that this Panel find Cylink is entitled to a license in the form submitted with its Closing Brief. As was discussed in some detail above, the Panel has found that to the extent RSA was obligated to offer to grant to Cylink any license under the MIT patent, it has done so both by providing the option to license in Exhibit 6 and by allowing PKP to offer the "LEMCOM type" license to Cylink. Therefore, the Panel does not find Cylink to be entitled to a license in the form submitted and this request is denied.

(ii) Breach of the Agreement by failure to deliver an MIT patent license: Cylink has requested that this Panel find a breach of the Agreement of Intent by RSA's failure to deliver a patent license. As the Panel has concluded that the required patent license was delivered as Exhibit 6, the Panel does not find a breach by RSA of the Agreement of Intent on this grounds.

(iii) A finding that RSA materially breached the Partnership Agreement by violating Article 6, ¶12 and an order entitling Caro-Kann to remedies pursuant to Article 9, Paragraph 5(d) and Cal.Corp. Code §15038: As stated above, the Panel does not find sufficient evidence to hold that either party has breached Article 6, Paragraph 12 or, at least to the extent a party has breached this paragraph, the other party has as well and the Panel would find the parties to be equally at fault for breach. Further, even if there was a material breach, the Panel does not find evidence of the notice with opportunity to cure required by the agreement. As a result, the request by Caro-Kann for an order entitling it to remedies pursuant to Article 9, Paragraph 5(d) and Cal. Corp.Code §15038 is denied.

As to other alleged breaches by RSA, the Panel does not at this time find it to be necessary to enter a finding of fault. However, if requested by Cylink or Caro-Kann, the Panel will consider the question of specific other alleged breaches and whether or not any such alleged breaches results in liability on the part of RSA.

(iv) A finding against RSA on all of the claims tendered by it: These claims and the panels findings will be discussed below.

(v) A finding that neither Cylink or Caro-Kann committed any material breach of the partnership agreement: Again, the Panel does not find sufficient evidence to hold that either party has breached Article 6, Paragraph 12 or, at least to the extent a party has breached this paragraph, the other party has as well and the Panel would find the parties to be equally at fault for breach. Further, even if there was a material breach, the Panel does not find evidence of the notice with opportunity to cure required by the agreement. Therefore, the Panel does not find any liability on the part of Cylink or Caro-Kann for any material breach of the partnership agreement.

(vi) A finding that Cylink and Caro-Kann are not liable for the duties or conduct of each other: The Panel finds that the corporate veil should not be pierced and that Cylink should not be liable for the liabilities of its subsidiary, Caro-Kann.

(vii) A decree dissolving Public Key Partners: The Panel has ordered Public Key Partners dissolved as of the date of this order consistent with the other orders and findings contained herein.

(b) Liability on the part of Cylink / Caro-Kann: The issues which the Panel has been requested to decide by RSA, and the Panel's findings and orders regarding those issues are as follows:

(i) A finding that Cylink has breached both the partnership agreement and its fiduciary duty owed to RSA and PKP and that RSA has been damaged thereby: As stated above, the Panel does not find sufficient evidence to hold that either party has breached Article 6, Paragraph 12 or, at least to the extent a party has breached this paragraph, the other party has as well and the Panel would find the parties to be equally at fault for breach. Further, even if there was a material breach, the Panel does not find evidence of the notice with opportunity to cure required by the agreement. Therefore, the Panel does not find any liability on the part of Cylink or Caro-Kann for any material breach of the partnership agreement.

As to other alleged breaches by Cylink or Caro-Kann, the Panel does not at this time find it to be necessary to enter a finding of fault. However, if requested by RSA, the Panel will consider the question of specific other alleged breaches and whether or not any such alleged breaches results in liability on the part of Cylink or Cara-Kann.

(ii) A finding that RSA has not breached its contractual or fiduciary obligations owed to Cylink or PKP and that RSA has the right to continue to license software that incorporates both the MIT and Stanford Patented Technology: Again, the Panel does not find sufficient evidence to hold that either party has breached Article 6, Paragraph 12 or, at least to the extent a party has breached this paragraph, the other party has as well and the Panel would find the parties to be equally at fault for breach. Further, even if there was a material breach, the Panel does not find evidence of the notice with opportunity to cure required by the agreement. Therefore, the Panel does not find any liability on the part of RSA for any material breach of the partnership agreement.

As stated above, the Panel does not find it to be necessary to reach a decision at this time regarding any other alleged breaches and will withhold any such findings unless specifically requested to Cylink or Caro-Kann to enter such findings.

With respect to RSA's right to license software that incorporates the MIT and Stanford patented technology, the Panel finds as follows:

a) Post-partnership formation licenses[1]: The Panel finds that RSA does not have the right to sublicense third-parties under the Stanford patents and has not had such right since entering into the partnership agreement on April 6, 1990. Therefore, after April 6, 1990, RSA has the the right to license its (RSA's) software to third-parties but does not have the right to license such third-parties under the Stanford patents. To the extent RSA provides code to third-parties which causes an infringement of a valid and enforceable claim of the Stanford patents, assuming the third party is not separately licensed under the Stanford patent, nothing in this order shall prevent Cylink from pursuing it rights under the Stanford patents against such third party. However, certain transactions between RSA and third-parties may be subject to the first sale doctrine. Nothing in this order is intended to remove application of that doctrine.

By way of example, if RSA provides a license of a single copy of its software, together with that copy of the software, to a third-party, the first sale doctrine will protect the third-party from any claim under the Stanford patents with respect to that individual copy. However, if RSA provides a license of its software, with right to prepare unlimited copies, to a third-party, the Panel finds that the first sale doctrine will not protect the third-party from suit on the Stanford patents.

b) Existing licensees: The Panel finds that RSA did have the right to sublicense the Stanford patents prior to April 6, 1990. Therefore, the Panel finds that any sublicense granted by RSA prior to April 6, 1990 is

[1] Nothing in this order shall effect the order previously issued by the Panel concerning Netscape.

not effected by this order and unless terminated pursuant to the terms of such license, is valid and allows the third-party rights under the Stanford patents as provided in the license.

(iii) A finding that Cylink is not contractually entitled and RSA is not contractually obligated to provide to Cylink a license to the MIT patent: As discussed above, except for the option to Cylink given in Exhibit 6, Cylink is not entitled to, contractually or under any of the other theories advanced by Cylink, a license to the MIT patent and RSA is, therefore, not obligated to provide any license to Cylink except the limited patent license afforded by Exhibit 6 if Cylink should choose to exercise its option.

(iv) A finding that the partnership shall be dissolved: The Panel has ordered Public Key Partners dissolved as of the date of this order consistent with the other orders and findings contained herein.

(v) An order that Cylink shall pay to RSA any and all of the costs and expenses incurred by RSA in the Arbitration including actual attorneys fees, as set forth in Article 12, Paragraph 6 of the Partnership Agreement: The Panel finds that neither party shall be entitled to an award of costs and expenses incurred to date in this arbitration.

IT IS SO ORDERED on this 6th day of September, 1995.

By: G. C. Limbach
George C. Limbach, Esq.
Chairman of the Arbitration Panel

By: Henry C. Bunsow
Henry C. Bunsow, Esq.
Member of the Arbitration Panel

By:
David R. Halvorson
Member of the Arbitration Panel